



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PEKKA MARJAMÄKI
JÄRJESTELMÄKAMERAN OHJAUS 3D-TUTKAN
KOORDINAATTITIEDON PERUSTEELLA

Diplomityö

Tarkastajat: yliopistonlehtori Juha T.
Tanttu ja diplomi-insinööri Juha
Niemi
Tarkastajat ja aihe hyväksytty
Tuotantotalouden ja rakentamisen
tiedekuntaneuvoston kokouksessa
9. marraskuuta 2016

TIIVISTELMÄ

PEKKA MARJAMÄKI: Järjestelmäkameran ohjaus 3D-tutkan koordinaattitiedon perusteella

Tampereen teknillinen yliopisto

Diplomityö, 59 sivua, 43 liitesivua

Marraskuu 2017

Johtamisen ja tietotekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Tietoverkkojen hallinta ja tietoturva

Tarkastajat: Yliopistonlehtori Juha T. Tanttu ja diplomi-insinööri Juha Niemi

Avainsanat: Tutka, WGS84, kääntöpää, TCP/IP, Pelco "D", RS-485

Tuulivoiman yleistyessä on kiinnitetty huomiota yhä enemmän myös sen ympäristövaikutuksiin. Tässä diplomityössä tutkittiin edullisen kääntöpään käyttöä osana järjestelmää, jolla tutkan antaman paikkatiedon perusteella järjestelmäkamera suunnataan kohteeseen. Järjestelmäkameralla otetun kuvan perusteella pyritään luokittelemaan kohde. Tutkimuskysymys oli, onko suuntaaminen mahdollista riittävällä tarkkuudella?

Diplomityössä perehdyttiin kirjallisuustutkimuksena eri koordinaattien esitystapoihin, esitystapojen välisiin muunnoksiin, korkeusjärjestelmään sekä paikkatietoon yleisesti. Suuressa roolissa oli myös eri tietoliikenneprotokolliin perehtyminen.

Diplomityön perusteella ei voida lopullisesti sanoa, onko kääntöpään suuntaaminen kohteeseen riittävällä tarkkuudella mahdollista. Käytetyn kääntöpään kääntymisnopeus on riittävä, mutta sen asettumistarkkuus ei mahdollista jatkuvaa toimintaa sellaisella toimintatavalla, joka lopullisessa toteutuksessa olisi tarkoituksenmukaista.

Tutkimuksessa havaittiin myös, että kameran ohjaukseen vaikuttaa merkittävästi myös tutkan lähettämän paikkatiedon tarkkuus. Nyt saatavilla ei ollut tutkan etäisyys- ja kulmaresoluution arvoja, joiden perusteella olisi voitu arvioida tutkan mittaustarkkuuden riittävyttä. Yksi virhelähde tutkassa saattaa olla myös menetelmä, jolla kohteen suunta- ja etäisyystiedot muunnetaan WGS84-koordinaatistoon.

Omat virheensä aiheuttavat myös tutkan ja kääntöpään omien sijaintipisteiden koordinaattien epätarkkuudet.

Tutkimuksen perusteella esitetään, että tulee joko sallia nykyisen kääntöpään suunnan toistuva kalibrointi, tai sen korvaaminen sellaisella tekniikalla, jonka asettumistarkkuus on huomattavasti parempi.

Tämä diplomityö tehtiin Hyötytuuli Oy:lle. Työ liittyy tuulisähkövoimaloiden turvallisuusjärjestelyihin, joissa lopullisena tavoitteena on pysäyttää tuulivoimala hallitusti, mikäli lintu on vaarassa törmätä sen roottorin lapoihin.

ABSTRACT

PEKKA MARJAMÄKI: SLR camera steering based on coordinate information from a 3D radar

Tampere University of Technology

Master of Science Thesis, 59 pages, 43 Appendix pages

November 2017

Master's Degree Programme in Management and Information Technology

Major: Network Management and Information Security

Examiners: University lecturer Juha T. Tanttu and Master of Science Juha Niemi

Keywords: Radar, WGS84, pan and tilt head, TCP/IP, Pelco "D", RS-485

Wind power has become increasingly popular. Environmental effects of the operations, such as birds colliding with the wind turbine, have gained more attention. This thesis studies the possibility to utilize a mid-price pan and tilt head as a part of the system where the SLR camera is steered based on the spatial data provided by the radar. The objective is to classify the target based on the picture taken by the SLR camera. Thus the research question is: can the steering be done with sufficient accuracy? The thesis was conducted for Hyötytuuli Oy. It is a part of security planning for stopping the wind turbine in a controlled manner when a bird is endangered to collide with the wind turbine.

The thesis includes an extensive literature review on different coordinate systems, the conversions of those dictions, and the altitude system and spatial data in general. Another major part of the thesis is to study different telecommunication protocols.

Sufficient accuracy of the head cannot be confirmed by the results of this thesis. The tested pan and tilt head has a sufficient rotation speed. However, it did not settle accurately enough to enable consistent operations, which would be adequate in the final implementation.

The spatial information provided by the radar has significant effect on the steering. No distance and angle resolution values of the radar were available for evaluating the sufficiency of its measurement accuracy. One source of error may also be a method to convert direction and distance data to WGS84-coordinates. The inaccuracies of the location coordinates of both radar and the head cause errors as well.

This study proposes that the head requires frequent calibration or that the head is replaced by a one more suitable for this purpose.

ALKUSANAT

Tämän diplomityön tekeminen, ja sitä edeltäneiden opintojen suorittaminen, oli raskas mutta opettava ja antoisa prosessi. Opiskelun loppua kohti kasaantuneet aikatauluongelmat opintojen, työn ja muiden kiireiden välillä, loivat työn loppuun saattamiselle huomattavia haasteita.

Tahdon kiittää perhettäni koko tämän noin kolmen vuoden projektin aikaisesta ymmärryksestä. Useamman viikonlopun ja arki-illan olen viettänyt suljettujen ovien takana. Vaimo, lapset ja kodinhoito on jäänyt vahvasti taka-alalle.

Suuri kiitos tämän työn valmistumisesta on annettava yliopistonlehtori Juha Tantulle. Juhan palaute ja ohjaus oli ensiarvoisen tärkeää. Kiitän myös tohtoriopintojaan suorittavaa Juha Niemeä teknisestä tuesta – erityisesti ohjelmoinnin osalta.

Lopuksi haluan vielä kiittää kaikkia kanssaopiskelijoita, joiden avulla jaksoin työpäivän jälkeen viettää aikaani luennoilla. Erityiskiitokset kohdistan Kristalle ja Soilille loppumetrieni kannustuksesta.

Loimaalla, 2.11.2017

Pekka Marjamäki

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	TUTKA	5
2.1	Tutkan historiaa	5
2.2	Tutkien luokittelu	6
2.3	Pulssitutkan ominaisuuksia	7
2.4	Taajuusmoduloidun kanta-aaltotutkan ominaisuuksia	10
2.5	Robinin 3D-tutka	11
3.	KÄÄNTÖPÄÄN PARAMETRIEN SELVITYS	12
3.1	Kääntöpään kääntymisnopeus	12
3.2	Kääntöpään asettumistarkkuus	13
4.	KOORDINAATISTOT JA KOORDINAATTIJÄRJESTELMÄT	16
4.1	Koordinaatisto	16
4.2	WGS84-koordinaattijärjestelmä	17
4.3	EUREF-FIN ja ETRS89-koordinaattijärjestelmät	18
4.4	Korkeusjärjestelmä	20
4.5	Koordinaatti- ja koordinaattijärjestelmämuunnokset	21
5.	KÄYTETYT TIETOLIIKENNEYHTEYDET JA –PROTOKOLLAT	24
5.1	Piiri- ja pakettikytkentäiset verkot	24
5.2	Protokollapino	25
5.3	Yhteydetön- ja yhteydellinen kommunikointi	26
5.4	TCP/ IP-protokollaperhe	27
5.5	RS-485 liitäntä ja Pelco ”D” protokolla	34
6.	KEHITETTY OHJELMISTO	37
6.1	Kääntöpään manuaalinen ohjaus	38
6.2	Tutkan simulointi	40
6.3	Ohjelmarunko laskentafunktioiden kehitykseen	41
6.4	Etäisyyden laskenta	43
6.5	Sivusuunnan laskenta	44
6.6	Korkeuskulman laskenta	52
6.7	Ennakkopisteen laskenta	53
6.8	Lähtötietojen tarkkuus	56
7.	YHTEENVETO JA JATKOKEHITYS	57
	LÄHTEET	59

LIITE A: USB – RS-485 -SOVITTIMEN ASETUKSET

LIITE B: PELCO “D” PROTOCOL MANUAL

LIITE C: Application programming interface

LIITE D: ASETTUMISTARKKUUDEN MITTAUSTULOKSET

LIITE E: PAN AND TILT HEADS, PT-1020 Medium Duty

- LIITE F: JHS 154 ETRS89 järjestelmään liittyvät karttaprojektiot, tasokoordinaatistot ja karttalehtijako, Liite 1: Projektiokaavat
- LIITE G: TECHNICAL DATA SHEET, ROBIN Specification Guide

KUVALUETTELO

Kuva 1.	<i>Tuulivoimaloiden määrän, tuotantokapasiteetin ja energian tuotannon kehitys suomessa vuosina 2013–2016. Perustuu lähteeseen (Suomen Tuulivoimayhdistys, 2017).</i>	1
Kuva 2.	<i>Karkea lohkokaavio toteutettavasta järjestelmästä.</i>	3
Kuva 3.	<i>Tutkan yksinkertaistettu lohkokaavio. Perustuu lähteeseen (Olli ja Lehto, 1999).</i>	5
Kuva 4.	<i>Yksiselitteinen mittausetäisyys ja jaksontakainen kaiku.</i>	9
Kuva 5.	<i>FMCW-tutkan lähetys- ja vastaanottosignaalien modulaatio. Perustuu lähteeseen (Olli ja Lehto, 1999).</i>	10
Kuva 6.	<i>Sivusuunnan kääntymisnopeuden mittausarvot ja kuvaaja.</i>	12
Kuva 7.	<i>Korkeussuunnan kääntymisnopeuden mittausarvot ja kuvaaja.</i>	13
Kuva 8.	<i>Kuvassa on esitetty ellipsoidikeskeinen kolmiulotteinen suorakulmainen koordinaatisto (x-, y- ja z-koordinaatit) ja maantieteellinen koordinaatisto (kulmat λ ja ϕ sekä korkeus h) sekä karttakoordinaatisto (X/P- ja Y/I.-koordinaatit). Perustuu lähteeseen (Laurila, 2012).</i>	16
Kuva 9.	<i>Maantieteellisten (λ, ϕ) sekä suorakulmaisten (x, y, z) koordinaattien vastaavuus. Perustuu lähteeseen (Laurila, 2012, s. 143).</i>	18
Kuva 10.	<i>UTM-projektion periaate sekä tasokoordinaatiston sijoittuminen muodostuneeseen projektiokaistaan. Lieriö voidaan oikaista tasoksi leikkaamalla se pituussuuntaan auki. Perustuu lähteeseen (Laurila, 2012, s. 146,151).</i>	19
Kuva 11.	<i>Mittakaavakerroin määräytyy projisoituneen matkan ja ellipsoidin pinnalla kuljetun matkan suhteesta. Perustuu lähteeseen (Laurila, 2012, s. 148,151).</i>	20
Kuva 12.	<i>Ellipsoidinen korkeus (h) on ortometrisen korkeuden (H) ja geoidin korkeuden (N) summa. Perustuu lähteeseen (Laurila, 2012, s. 166).</i>	21
Kuva 13.	<i>Tietoliikenneyhteydet ja niillä siirrettävä informaatio.</i>	24
Kuva 14.	<i>TCP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 244).</i>	29
Kuva 15.	<i>UDP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 164).</i>	30
Kuva 16.	<i>IP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 48).</i>	31
Kuva 17.	<i>TCP- ja UDP-pakettien kapselointi IP-paketin sisään. perustuu lähteeseen (Stevens, 1993, s. 163,244).</i>	33
Kuva 18.	<i>Datan siirtyminen lähettävältä sovellukselta vastaanottavalle sovellukselle. Perustuu lähteeseen (Stevens, 1993).</i>	33
Kuva 19.	<i>Pelco ”D” –protokollan sanomarakenne. Tavun 4 arvo on esitetty binäärisenä, muut tavut ovat heksadesimaalimuodossa.</i>	35
Kuva 20.	<i>Modulo 256 tarkastussumman laskenta binäärisenä.</i>	36
Kuva 21.	<i>Eri tietoliikenneyhteyksillä lähetettävät tiedot.</i>	37
Kuva 22.	<i>Bittitason looginen OR-operaatio.</i>	39

Kuva 23.	<i>Bittitason loogiset NOT- ja AND-operaatiot.....</i>	<i>40</i>
Kuva 24.	<i>Kolmiulotteisen XYZ-koordinaatiston XY-projektiotaso.</i>	<i>45</i>
Kuva 25.	<i>Suuntakulma kun kohde sijaitsee eri neljänneksissä.</i>	<i>46</i>
Kuva 26.	<i>PI-tasokoordinaatisto.</i>	<i>48</i>
Kuva 27.	<i>Suuntakulman laskeminen PI-tasokoordinaatistossa.</i>	<i>48</i>
Kuva 28.	<i>Maapallon kaarevuuden tarkastelua.</i>	<i>52</i>
Kuva 29.	<i>Kohteen korkeuden ja korkeuskulman laskentaperiaate.</i>	<i>53</i>

TAULUKKOLUETTELO

Taulukko 1. <i>ISO/OSI-mallin ja TCP/IP-mallien kerrosten vastaavuus. Perustuu lähteeseen (Zdonik ja Katz, 2014).</i>	26
Taulukko 2. <i>Esimerkkejä TCP/IP-mallin eri kerroksilla toimivista protokollista. Perustuu lähteeseen (Zdonik ja Katz, 2014).</i>	27
Taulukko 3. <i>TCP-paketin eri kenttien sisältö ja merkitys. Perustuu lähteeseen (Stevens, 1993, s. 244–246,250,317).</i>	29
Taulukko 4. <i>UDP-paketin eri kenttien sisältö ja merkitys. Perustuu lähteeseen (Stevens, 1993, s. 244–246,250,317).</i>	31
Taulukko 5. <i>IP-paketin eri kenttien koko ja käyttötarkoitus. Perustuu lähteeseen (Stevens, 1993, s. 25–26,48–51,169).</i>	32
Taulukko 6. <i>XY-tasosta lasketun suuntakulman laskentakaavan korjaus.</i>	47
Taulukko 7. <i>PI-tasosta lasketun suuntakulman laskentakaavan korjaus.</i>	49
Taulukko 8. <i>Ennakkopisteen iteroinnin kehittyminen.</i>	54

LYHENTEET JA MERKINNÄT

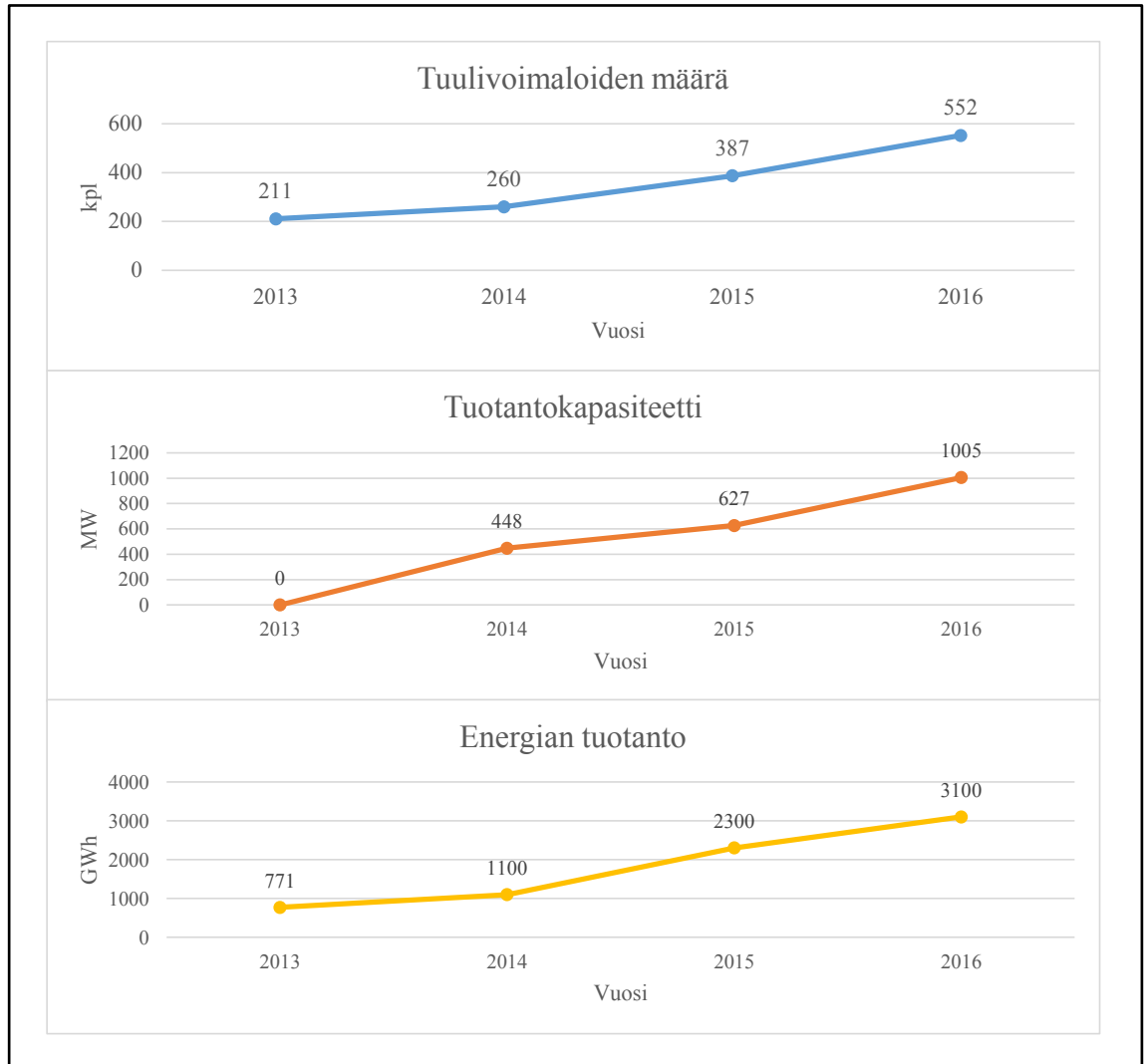
3D	Kolmiulotteinen, (eng. 3-dimensional , three dimensional).
ARPANET	Advanced Research Projects Agency Network .
ACK	Acknowledgement . TCP-segmentin lippu, joka kertoo, että kuittausnumero (acknowledgement number) on asetettu.
AWG	American wire gauge . Amerikkalainen johtimen poikkipinnan mitta. Esimerkiksi AWG 24 vastaa halkaisijaltaan 0,511 mm, eli 0,205 mm ² johdinta.
b	Ympyrän janteen puolikas.
B	Bird . Lintu, linnun (kohteen) sijainti kaaviokuvissa.
B_T	FMCW-tutkan lähetystaajuuden kokonaismuutos.
C	Camera . Kameran sijainti kaaviokuvissa.
c	Valon nopeus tyhjiössä, n. 300 000 km/s.
cf	Rajauskerroin (eng. cropp factor).
CW	Continuous wave . Jatkuva lähete, kantaalto.
DF	Don't fragment . IP-paketin lippu, joka estää paketin fragmentoitumisen.
e	Ellipsoidin ensimmäinen eksentrisyys.
E	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - East (itä). Käytetään mm. maantieteellisten pituuskulmien etuliitteenä - Easting UTM-projektion suorakulmaisen koordinaatiston pohjoiskoordinaatin etuliite - kahden pisteen välinen suora etäisyys.
ETRS89	European Terrestrial Reference System 1989 .
ETRS-TM35FIN	European Terrestrial Reference System – Transverse Mercator 35 – Finland . Suomen kansallinen UTM-projektio.
EUREF	European Reference Frame .
EUREF-FIN	European Reference Frame – Finland . ETRS89-koordinaattijärjestelmän Suomen kansallinen realisaatio.
f	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - taajuus, (eng. frequency) - polttoväli (eng. focal length).
f_p	Tutkan pulssintoistotaajuus.
f_R	Tutkan vastaanottotaajuus.
f_T	Tutkan lähetystaajuus.
FIN	Finished . TCP-segmentin lippu, joka kertoo, että lähettäjä on lopettanut datan lähettämisen.
FM	Frequency modulation . Taajuusmodulaatio.
FMCW	Frequency modulated continuous wave , taajuusmoduloitu kantaalto.
GPS	Global Positioning System . Maailmanlaajuinen satelliittipaikannusjärjestelmä, jota ylläpitää Yhdysvaltain puolustusministeriö.
GRS80	Geodetic Reference System 1980 .
h	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - pisteen korkeus vertausellipsoidin pinnasta. Ellipsoidinen korkeus - ympyrän janteen kohtisuora etäisyys ympyrän keskipisteestä.
H	Pisteen korkeus geoidin pinnasta. Ortometrinen korkeus.

<i>i</i>	Itäkoordinaatin arvo.
I	Itä. Käytetään mm. maantieteellisten pituuskulmien etuliitteenä.
ITRS	I nternational T errestrial R eference S ystem.
IP	I nternet P rotocol. Internet protokolla. OSI-mallin verkkokerroksen tai TCP/IP-mallin internetkerroksen pakettipohjainen protokolla joka huolehtii mm. hyötykuorman reitityksestä IP-verkoissa.
ISO	I nternational O rganization for S tandardization. Kansainvälinen standardisoimisjärjestö.
JUHTA	J ulkisen h allinnon t ietohallinnon neuvottelukunta.
JHS	J ulkisen h allinnon s uositus.
<i>K</i>	K urssi, purjehdussuunta.
kbps	K ilobits p er s econd. Kilobittiä sekunnissa.
L	Länsi. Käytetään mm. maantieteellisten pituuskulmien etuliitteenä.
lat	Latitudi (eng. l atitude), maantieteellisten koordinaattien leveyskulma.
lon	Longitudi (eng. l ongitude), maantieteellisten koordinaattien pituuskulma.
L/V-kytkin	Lähetys- / vastaanottokytkin.
<i>m, m</i>	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - metri. SI-järjestelmän pituuden perusyksikkö - UTM-projektion mittakaavakerroin (<i>m</i>).
MF	M ore f ragment. IP-paketin lippu, joka kertoo, että segmentti on fragmentoitu ja alipaketteja on vielä tulossa vastaanottoon.
Mbps	M egabits p er s econd. Megabittiä sekunnissa.
MSS	M aximum S egmen S ize. TCP-segmentin maksimikoko.
<i>N, N</i>	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - geoidin korkeus vertausellipsoidin pinnasta (<i>N</i>) - North (pohjoinen). Käytetään mm. maantieteellisten leveyskulmien etuliitteenä - Northing. UTM-projektion suorakulmaisen koordinaatiston pohjoiskoordinaatin etuliite.
<i>N_R</i>	Poikittaiskaarevuussäde.
OSI	O pen S ystems I nterconnection. OSI-malli kuvaa tiedonsiirtoprotokollien yhdistelmää seitsemällä eri palvelukerroksella.
<i>p</i>	Pohjoiskoordinaatin arvo.
<i>P, P</i>	Asiayhteydestä riippuen: <ul style="list-style-type: none"> - signaalin teho, (eng power) (<i>P</i>) - Pohjoinen. Käytetään mm. maantieteellisten leveyskulmien etuliitteenä.
PDU	P rotocol D ata U nit. Datapaketti.
PSH	P ush. TCP-segmentin lippu, joka kertoo vastaanottajalle, että data on välitettävä sovellukselle mahdollisimman nopeasti.
<i>r</i>	R adius. Ympyrän tai pallon säde.
<i>R</i>	Tutkan mittaama suora etäisyys kohteeseen.
<i>R_{min}</i>	Tutkan minimimittausetäisyys.
<i>R_{max}</i>	Tutkan yksikäsitteinen mittausetäisyys.
RST	R eset. TCP-segmentin lippu, joka kertoo, että yhteys on alustettava.
RS-485	R ecommended S tandard- 485 . Differentiaalinen (balansoitu) sarjaliikenneväylä. Käyttää vuorosuuntaista liikennöintiä väylässä, jossa

	voi olla max. 32 laitetta. Siirtotienä parikaapeli, jonka maksimi pituus n. 1200 m ja siirtonopeus 100 kbps ... 35 Mbps.
S	South (etelä). Käytetään mm. maantieteellisten leveyskulmien etuliitteenä.
s	Ympyrän janteen korkeus.
SYN	Synchronize . TCP-segmentin lippu, joka kertoo, että järjestysnumerialaskurit on synkronoitava yhteyden alustamiseksi.
T_m	FMCW-tutkan lähetystaajuuden pyyhkäisy aika.
T_p	Pulssintoistoväli.
t	Aika joka kuluu tutkan lähettimen lähetyspulssin lähetys hetkestä sen aiheuttaman kaikupulssin palaamiseen vastaanottimeen.
TCP	Transmission Control Protocol . OSI- sekä TCP/IP-mallien kuljetuskerroksen protokolla, jolla muodostetaan (neuvotellaan) luotettava (datapaketin perille meno varmistetaan), yhteydellinen tietoliikenneyhteys kahden pisteen välille.
TOS	Type of Service . IP-paketin kenttä, joka kertoo minkä tyyppinen palvelu on kyseessä.
UDP	User Datagram Protocol . OSI- sekä TCP/IP-mallien kuljetuskerroksen protokolla, jolla luodaan yhteydetön tietoliikenneyhteys kahden pisteen välille. UDP-yhteydessä ei datapaketin perillemeno varmisteta.
URG	Urgent . TCP-segmentin lippu, joka kertoo, että tärkeän datan osoitin (urgent pointer) on asetettu.
USB	Universal Serial Bus . Tietokoneen yleiskäyttöinen sarjaliikenneväylä.
UTM	Universal Transverse Mercator . Maailmanlaajuisesti käytetty kartta-projektio, joka on oikeakulmainen, poikittaisasentoinen ja leikkaava lieriöprojektio.
W	West (länsi). Käytetään mm. maantieteellisten pituuskulmien etuliitteenä.
WGS84	World Geodetic System 1984 . Maailmanlaajuinen koordinaattijärjestelmä, joka on käytössä mm. GPS-järjestelmässä.
X	Suorakulmaisen koordinaatiston X -akseli.
x	Suorakulmaisen koordinaatiston x -koordinaatti.
Y	Suorakulmaisen koordinaatiston Y -akseli.
y	Suorakulmaisen koordinaatiston y -koordinaatti.
Z	Suorakulmaisen koordinaatiston Z -akseli.
z	Suorakulmaisen koordinaatiston z -koordinaatti.
Δt	Ajan muutos.
Δf	Taajuuden muutos.
Δx	Kahden x-koordinaatin välinen ero.
Δy	Kahden y-koordinaatin välinen ero.
Δz	Kahden z-koordinaatin välinen ero.
δ_A	Tutkan kulmaerottelukyky.
δ_R	Tutkan etäisyydenerottelukyky.
φ	Maantieteellisten koordinaattien leveyskulma, latitudi.
λ	Maantieteellisten koordinaattien pituuskulma, longitudi.
Ω	Resistanssin ja impedanssin yksikkö.
τ	Pulssitutkan lähetyspulssin pituus.
θ_{3dB}	Antennin keilanleveys mitattuna 3 dB:n pisteistä.

1. JOHDANTO

Tuulivoiman rakentaminen on voimakkaassa kasvussa Suomessa. Kuvassa 1 on esitetty tuulivoiman kehitystä Suomessa.



Kuva 1. Tuulivoimaloiden määrän, tuotantokapasiteetin ja energian tuotannon kehitys suomessa vuosina 2013–2016. Perustuu lähteeseen (Suomen Tuulivoimayhdistys, 2017).

Tuulivoimaloiden määrän kasvaessa myös niiden vaikutukset ympäristöön kasvavat. Tuulivoimaloiden merkittävimmät haitalliset vaikutukset syntyvät muun muassa lintujen törmäyksistä tuulivoimaloiden rakenteisiin ja voimalinjoihin. Voimaloilla on myös lintuja karkottava vaikutus, joka merialueilla ulottuu jopa 2-4 km etäisyydelle voimalasta.

Tuulivoimalat voivat siis vaikuttaa esimerkiksi lintuparviin lentoreitteihin. (Pihlaja, 2012, s. 1; Motiva OY, 2015)

Käytönaikaisella seurannalla ja esimerkiksi voimaloiden hetkellisellä pysäyttämällä tarpeen vaatiessa, voidaan vähentää lintujen kuolleisuutta merkittävästi. Tämä vaatii aktiivisia seurantamenetelmiä, kuten tutkaseurantaa. (Pihlaja, 2012, s. 1)

Lintujen törmäämiset tuulivoimaloihin ovat suorassa suhteessa lintujen määrään tuulivoimaloiden lähellä. Pikkulintuja ja yöllä muuttavia lintuja kuolee suhteessa vähemmän törmäyksissä tuulivoimaloihin, kuin törmäyksissä rakennuksiin ja mastoihin. Isojen lintujen, kuten petolintujen kohdalla, tilanne on kuitenkin päinvastainen. (Motiva OY, 2015)

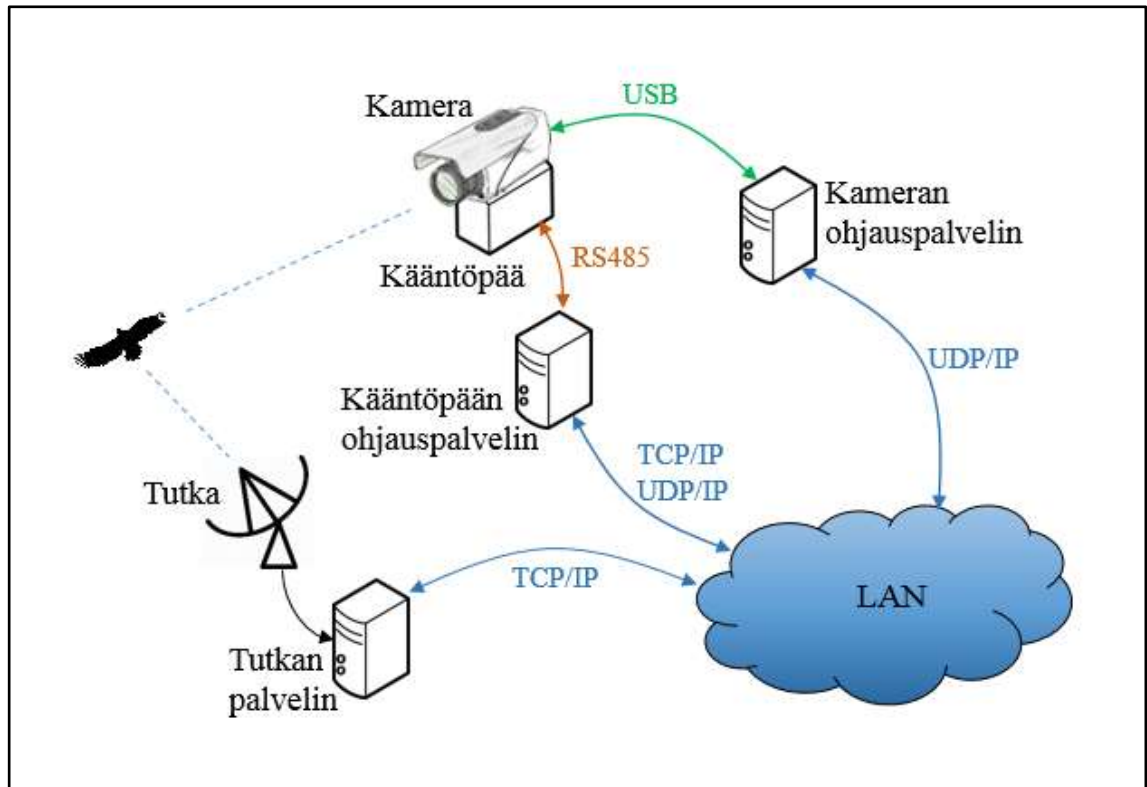
Yksi vaikutuksista, joita pyritään minimoimaan, on lintujen törmääminen voimaloiden pyöriviin lapoihin. Törmäämisten estämiseksi kehitetään erilaisia menetelmiä lintujen seuraamiseen sekä niiden käyttäytymisen ymmärtämiseksi. Käytettyjä menetelmiä ovat perinteisen näköhavainnoinnin lisäksi erilaiset akustiset- ja optiset järjestelmät sekä jo aikaisemmin mainitut tutkat. (Komatineni, 2011, ss. 1–2)

Tämä diplomityö liittyy kuvan 2 järjestelmään, jota kehitetään merialueella sijaitsevan offshore-tuulivoimalan läheisyydessä liikkuvien lintujen tarkkailuun ja tunnistukseen.

Tutkittava järjestelmä koostuu kokonaisuudessaan kuudesta päälohkosta:

- 3D-tutka (kolmiulotteinen tutka)
- 3D-tutkan palvelin
- kääntöpää järjestelmäkameran suuntaamiseksi
- järjestelmäkamera (polttoväli $f = 300 / 600$ mm ja rajauskero $cf = 1,62$)
- järjestelmäkameran ohjauspalvelin
- kääntöpään ohjauspalvelin tutkadatan käsittelyyn ja kääntöpään sekä kameran ohjaamiseen

Järjestelmäkameran polttovälin ja rajauskertoimen perusteella objektiivin avauskulma 300 mm polttovälillä on vaakasuunnassa noin $4,2^\circ$ ja pystysuunnassa noin $2,8^\circ$. Vastavasti 600 mm polttovälillä arvot puolittuvat ja ovat siis vaakasuunnassa $2,1^\circ$ ja pystysuunnassa $1,4^\circ$. Avauskulmat muodostavat vaatimuksen kääntöpään suuntauksen tarkkuudelle, koska kohde pitää saada kameran näkökenttään.



Kuva 2. Karkea lohkokaavio toteutettavasta järjestelmästä.

Tutkimuskysymys on, kyetäänkö 3D-tutkan informaation perusteella suuntaamaan järjestelmäkamera lentävään kohteeseen riittävällä tarkkuudella ja tuottamaan kameralle riittävän tarkka etäisyystieto. Suuntauksen tarkkuus on riittävä, mikäli kohde saadaan osumaan kameran näkökenttään kuvan ottamishetkellä.

Tutkimuskysymys jakaantuu neljään osakokonaisuuteen, jotka omalta osaltaan vaikuttavat lopputulokseen: suuntauksen tarkkuus, tutkan antaman paikkatiedon tarkkuus, kohteen liiketilän muuttuminen ja laskennan tarkkuus.

Tutkimuskysymyksen selvittämiseksi tehtiin testausta varten tietokoneohjelmisto, joka tutkan informaation perusteella laskee ennakkopisteen kohteen sijainnille. Ennakkopisteen laskennassa huomioidaan kohteen liike kääntöpään suuntaamisen aikana. Ohjelma myös ohjaa kääntöpään kääntymistä niin, että se osoittaa edellä mainittuun ennakkopisteeseen. Suuntauksen valmistuttua ohjelma lähettää kohteen etäisyystiedon kameran ohjauspalvelimelle.

Luvussa 2 esitellään lyhyesti tutkatekniikan perusteita sekä 3D-tutkan käyttämiä tekniikoita. Luvussa 3 esitellään valitun kääntöpään ominaisuuksia sekä siihen liittyviä mittauksia, joita tarvittiin kääntymistä ohjaavassa ohjelmassa tarvittavien parametrien selvittämiseksi.

Ohjelmisto joutuu muuntamaan WGS84-koordinaatiston (World Geodetic System 1984) paikkatiedon suuntatiedoksi kääntöpäälle (sivu- ja korkeuskulma) sekä etäisyystiedoksi

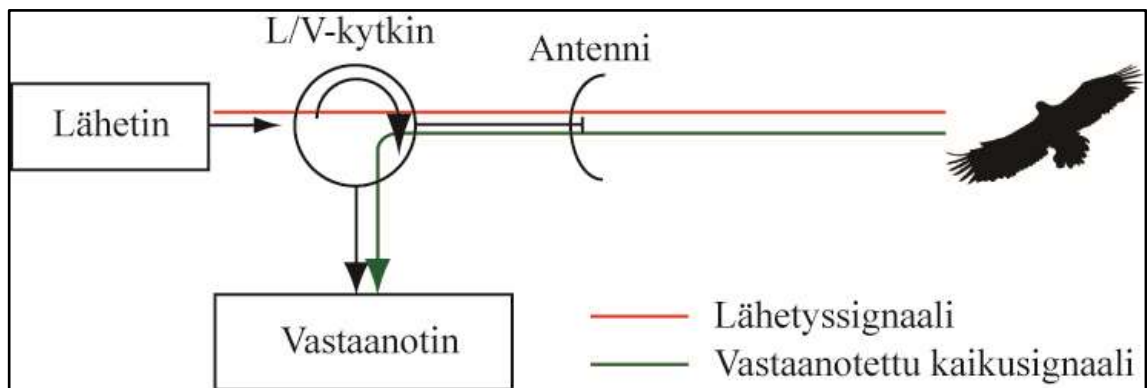
järjestelmäkameralle. Luvussa 4 käsitellään koordinaatistoja ja koordinaattijärjestelmiä. Järjestelmässä käytetään TCP (Transmission Control Protocol)- ja UDP (User Datagram Protocol)-protokollia IP-verkkoyhteydellä sekä Pelco ”D” -protokollaa RS-485 -yhteydellä. Luvussa 5 esitellään edellä mainitut protokollat ja yhteydet.

Luvussa 6 käsitellään ohjelmistokehitystä ja sekä laskennassa käytettyjä matemaattisia perusteita. Yhteenveto, päätelmät ja jatkokehitysideat ja -mahdollisuudet esitetään luvussa 7.

2. TUTKA

Tutka lähettää signaalin (radiotaajuisia, sähkömagneettisia aaltoja) haluttuun suuntaan ja vastaanottaa sekä ilmaisee kohteesta heijastuneen signaalin. (Richards *ym.*, 2008, s. 4)

Kuvassa 3 on esitetty tutkan yksinkertaistettu lohkokaavio. *Lähetysignaali* tuotetaan lähettimessä, josta signaali johdetaan siirtolinjan kautta L/V-kytkimelle (lähetys-/vastaanottokytkin). L/V-kytkimen tehtävä on ohjata lähetysignaali antenniin ja palaava signaali vastaanottimelle. Antenni sovittaa siirtolinjan vapaaseen tilaan, jolloin siirtymässä syntyy mahdollisimman pieni tehohäviö. Lähetysignaali etenee ilmakehässä valonnopeudella ja osuttuaan kohteeseen heijastuu kohteesta takaisin *kaikusignaali*. Matkalla lähetetty sekä palaava signaali vaimenevat ja niihin summautuu erilaisia häiriöitä. Heijastunut signaali vastaanotetaan antennilla ja L/V-kytkin ohjaa signaalin vastaanottimelle. Vastaanottimessa tyypillisiä operaatioita ovat signaalin siirtäminen välitaajuudelle, signaalin ilmaisu, digitalisointi sekä signaalinkäsittely. (Olli ja Lehto, 1999, ss. 22–23, 28–31; Richards *ym.*, 2008, s. 4)



Kuva 3. Tutkan yksinkertaistettu lohkokaavio. Perustuu lähteeseen (Olli ja Lehto, 1999).

2.1 Tutkan historiaa

Cristian Hülsmeuer patentoi 22.9.1904 ”telemobiloskoopin”. Telemobiloskoopia oli tarkoitus käyttää laivoissa törmäyksenestolaitteena, mutta se ei tuolloin saanut yleistä suosiota. Tätä laitetta pidetään kuitenkin ensimmäisenä tutkana. Telemobiloskoopin kantama oli vain muutama kilometri ja se ilmaisi törmäyskurssilla olevan aluksen hälytyskellon avulla. Vuoden 1904 jälkeen tutkateknikka on kehittynyt huimasti. 1920-luvulla pulssitekniikkaa kokeiltiin ensimmäisen kerran ja kanta-aaltotutkalla tehtiin hajanaisia kokeita. 1930-luvulla alettiin tutkaa kehittää järjestelmällisesti. Vuonna 1940 brittiläiset Randall

ja Boot keksivät ontelomagnetronin, joka mahdollisti suurien pulssitehojen käytön suurilla taajuuksilla. Samana vuonna ontelomagnetroniin perustuva tutka ylsi jo 10 GHz:n lähetystaajuuteen. (Olli ja Lehto, 1999, ss. 10–11)

Toisen maailmansodan jälkeen kiinnitettiin huomiota tutkan signaalin käsittelyyn. Tämä oli seurausta sodan aikana mahdolliseksi todetusta tutkahäirinnästä, -harhautuksesta ja -tiedustelusta. *Liikkuvan maalin ilmaisu* ja *pulssikompressio* tulivat osaksi tutkatekniikkaa. 1960- ja 1970-luvuilla kehittyi antennitekniikka ja ensimmäiset vaiheohjatut antenniryhmät kehitettiin. Prosessoritekniikan kehittyessä myös digitaalinen signaalinkäsittely tuli mukaan tutkatekniikkaan. (Olli ja Lehto, 1999, ss. 11–12)

Tutkatekniikka on erittäin laaja tekniikan alue pitäen sisällään lähetin-, vastaanotin- ja antennitekniikkaa sekä suurjännite- ja mikroaaltotekniikkaa. Lisäksi tutkaan liittyy oleellisesti signaalin käsittely sekä näyttölaite, joissa myös on runsaasti omia erityisalueitaan.

2.2 Tutkien luokittelu

Tutkia luokitellaan useiden eri ominaisuuksien perusteella. Luokittelua tehdään esimerkiksi tutkan rakenteen, kaikupulssin syntymekanismin, lähettimen ja vastaanottimen signaalin aaltomuodon tai ilmaisuperiaatteen perusteella.

Luokittelu *yksi-, kaksi- tai monipaikkatutkiin* perustuu tutkan antennirakenteeseen tai lähettimien ja vastaanottimien maantieteelliseen sijoitteluun. Yksipaikkatutkassa käytetään samaa antennia sekä lähetyksessä että vastaanotossa. Monipaikkatutkassa taas lähettimellä ja vastaanottimella on oma antenni. Tätä ajattelua voidaan laajentaa koskemaan myös lähettimen ja vastaanottimen sijaintia. Yksipaikkatutkassa lähetin ja vastaanotin ovat samassa maantieteellisessä sijainnissa. Kaksipaikkatutkan tapauksessa, ne voidaan sijoittaa hyvinkin etäälle toisistaan. Ehkä yleisin esimerkki kaksipaikkatutkasta on ohjuksen puoliaktiivisessa tutkaohjauksessa. Maali ”valaistaan” tutkalla ja ohjus seuraa maalista heijastuvaa signaalia. Monipaikkatutkassa voidaan kokonaisuuden muodostavat lähettimet ja vastaanottimet sijoittaa laajalle alueelle. Tällaista järjestelmää on vaikea häiritä. Monipaikkatutkalla myös mittausetäisyyttä on mahdollista kasvattaa. Koska tutkan lähete etenee normaalisti suoraviivaisesti, maapallon kaarevuus rajoittaa mittausetäisyyttä. Monipaikkatutkassa signaalin ei tarvitse palata lähetyspisteeseen, mikä mahdollistaa niin sanotun horisontin taakse mittaamisen. (Olli ja Lehto, 1999, ss. 17–18)

Ensiö- ja toisiotutkat erotellaan toisistaan sen perusteella, miten kaikupulssi syntyy. Ensiötutkassa lähetetty signaali heijastuu kohteesta ja palaa vastaanottimeen. Toisiotutkassa lähetetty signaali vastaanotetaan kohteen vastaanottimella ja siihen lähetetään vastaus lähettimellä. Toisiotutka vastaanottaa kohteen vastauksen. Vastauksessa välitetään usein lisätietoja kuten lennon numero tai koneen tyyppi. (Olli ja Lehto, 1999, s. 18)

Kantaaalto- ja pulssitutkat erotellaan toisistaan signaalin aaltomuodon perusteella. Yleisesti tutkan lähetyssignaali jaetaan kahteen luokkaan: jatkuvaan signaaliin (kantaaalto) sekä pulssimuotoiseen signaaliin. Kantaaaltotutkassa lähetin lähettää ja vastaanotin vastaanottaa signaalia jatkuvatoimisesti. Kantaaaltotutkat ovat usein kaksipaikkatutkia (erilliset antennit), jolloin saadaan parempi erotus lähetyks- ja vastaanottosignaalien välillä. Koska lähetyks- ja vastaanottosignaalien erotus ei ole kuitenkaan täydellinen, rajoittaa se lähetystehoa ja siten myös mittausepäilyä. Maalin säteittäinen nopeus tutkaa kohti saadaan laskettua dopplertaajuuden perusteella. Koska kantaaaltotutkan lähete on jatkuva, pitää kulkuajan mittaamista varten (etäisyyden mittausta) signaaliin liittää jokin ”merkki”. Tämä tehdään yleensä taajuusmoduloinnilla, missä signaalin taajuutta muutetaan ajan funktiona. Puhuttaessa kantaaaltotutkista käytetään usein englanninkielisiä lyhenteitä CW (continuous wave)- tai FMCW (frequency modulated continuous wave)-tutka. (Olli ja Lehto, 1999, s. 22; Richards *ym.*, 2008, s. 20)

Pulssitutkassa lähetetään suuritaajuisia signaalia hyvin lyhyen ajan. Suuritaajuuspulssin *kulku-aika* mitataan ja sen perusteella määritetään kohteen etäisyys. Pulssin lähetyksen aikana antenni on kytketty L/V-kytkimen avulla lähettimeen, jolloin vastaanotin ei voi vastaanottaa. Pulssin lähetyksen jälkeen L/V-kytkin kytkee antennin vastaanottimeen ja alkaa *kuuntelujakso*. Kuuntelujakson aikana vastaanotetut heijastuneet kaikusignaalit ilmaistaan ja prosessoidaan. Lähetyspulssin pituus, τ , ja kuuntelujakson pituus määrittävät tutkan *pulssintoistovälin*, T_p , sitä kautta myös *pulssintoistotaajuuden*, f_p , (Olli ja Lehto, 1999, ss. 28, 35; Richards *ym.*, 2008, s. 20)

Ero *koherentin ja epäkoherentin* tutkan välillä muodostuu vastaanottimen ilmaisimessa. Epäkoherentti vastaanotin kykenee ilmaisemaan vain signaalin amplitudin. Koherentti vastaanotin taas kykenee ilmaisemaan sekä signaalin amplitudin että vaihemuutoksen. Vaiheilmaisus vaatii toimiakseen vaihetarkan referenssisignaalin, jonka tuottaminen joissakin tapauksissa saattaa olla haasteellista. Koherentti vastaanotin mahdollistaa mm. liikuvien maalien ilmaisun, koska vaiheilmaisimella saadaan näkyviin etäisyyden muutoksen ja dopplertaajuuden aiheuttama vaihe-ero. (Olli ja Lehto, 1999, ss. 95, 154–155; Richards *ym.*, 2008, s. 23)

2.3 Pulssitutkan ominaisuuksia

Kuten edellä on todettu, pulssitutkan lähetin lähettää suuritaajuuspulssin määräväleihin ja kuuntelee sen jälkeen kuuntelujakson ajan lähetteen tuottamia kaukupulseja. Kohteen etäisyys R voidaan laskea pulssin edestakaisesta kulkuajasta t , kaavalla (Olli ja Lehto, 1999, s. 28; Richards *ym.*, 2008, s. 22)

$$R = \frac{t \cdot c}{2} \tag{1}$$

missä c on valon nopeus tyhjiössä, noin $3 \cdot 10^8 \frac{m}{s}$.

Lähetysten aikana vastaanotin on erotettu antennista L/V-kytkimen avulla. Näin ollen lähetysten aikana ei voida vastaanottaa kaikusignaalia. Tästä seuraa, että tutkalla on *minimimittausetäisyys*, R_{min} , joka voidaan laskea pulssinpituuksella kaavalla (Olli ja Lehto, 1999, ss. 41–42; Richards *ym.*, 2008, s. 502)

$$R_{min} = \frac{\tau \cdot c}{2} \quad (2)$$

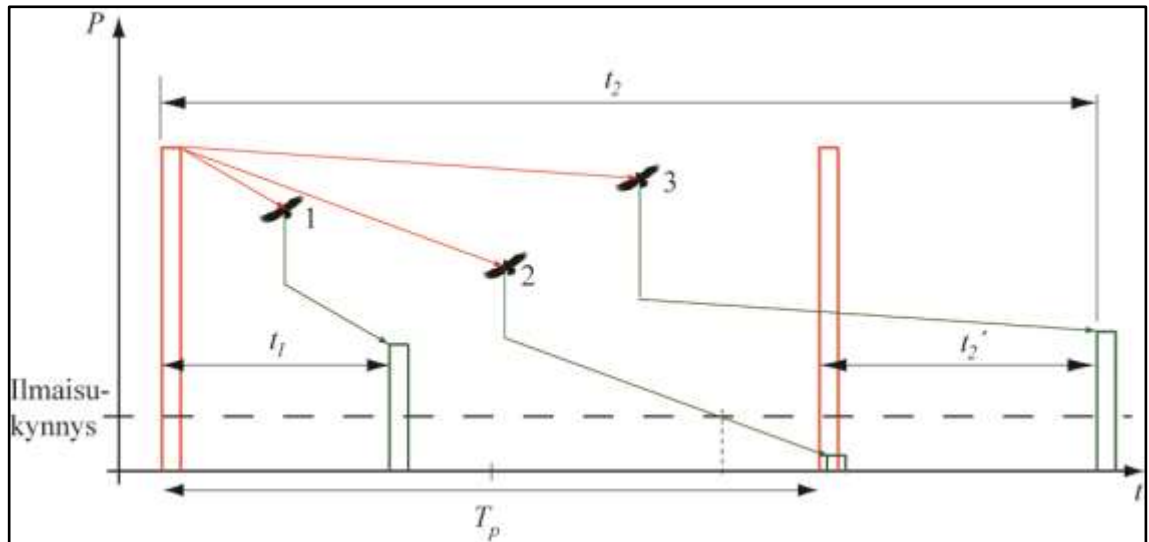
Käytännössä minimimittausetäisyyden arvo on usein suurempi, koska esimerkiksi L/V-kytkimen tilan vaihtuminen ja mahdollinen vastaanottimen elpyminen lähetyspulssin vuototehosta vaatii myös aikaa.

Pulssitutkan *maksimimittausetäisyyteen* vaikuttaa useat eri tekijät kuten, lähetysteho, antennivahvistus, lähteen aallonpituus, kohteen tutkapoikkipinta-ala, vastaanottimen herkkyys ja niin edelleen. Osaa vaikutuksista on helppo arvioida, mutta esimerkiksi ilma-kehan vaimennuksen, sadevaimennuksen ja radioaaltojen taipumisen vaikutuksien arvioiminen on erittäin haasteellista. Tutka pyritään suunnittelemaan niin, että edellä mainitut vaikeasti määriteltävät tekijät rajoittavat maksimimittausetäisyyden pienemmäksi kuin *yksikäsitteinen mittausetäisyys*. (Olli ja Lehto, 1999, ss. 41, 50, 53)

Yksikäsitteisen mittausetäisyyden määrittää tutkan kuuntelujakson pituus. Edellisen kappaleen perusteella oletetaan, että kohteesta saatu kaikusignaali tulee vastaanottimeen ennen uuden lähetyspulssin lähettämistä. Pulssintoistoväli valitaan niin pitkäksi, että yksikäsitteisen mittausetäisyyttä kauempaa tulevan paluusignaalin teho ei ylitä ilmaisukynnystä. Yksikäsitteinen mittausetäisyys, R_{max} , voidaan laskea pulssintoistovälistä, T_p , kaavalla (Olli ja Lehto, 1999, s. 43; Richards *ym.*, 2008, ss. 22–23, 502)

$$R_{max} = \frac{T_p \cdot c}{2} = \frac{c}{2 \cdot f_p} \quad (3)$$

Jos jostakin syystä yksiselitteistä mittausetäisyyttä kauempaa saadaan ilmaisukynnyksen ylittävä kaiku, sen etäisyys on väärä, koska etäisyys lasketaan viimeisen lähetyspulssin lähetyshetkestä. Tällaista kaikua kutsutaan *jaksontakaiseksi kaiuksi* (eng. *second time around echo*). (Olli ja Lehto, 1999, s. 43; Richards *ym.*, 2008, s. 22)



Kuva 4. Yksiselitteinen mittausetäisyys ja jaksontakainen kaiku.

Kuvassa 4 on esitetty yksinkertaistettu kuvaus lähetyspulssin (punainen) ja sen aiheuttaman kaikupulssin (vihreä) kulkuajasta sekä vaimenemisesta kolmessa eri tapauksessa. Tapauksessa 1 edestakainen kulku-aika on t_1 , ja kaikupulssin teho on vastaanottimeen saapuessaan yli ilmaisukynnyksen. Maalin etäisyys voidaan laskea kaavan 1 avulla. Tapauksessa 2 kaikupulssi on heijastunut kohteesta, joka on tutkan yksikäsitteisen mittausetäisyyden ulkopuolella. Edestakainen kulku-aika on tällöin lyhyempi kuin pulssintoistoväli. Kuten edellä todettiin, on tutka suunniteltu niin, että tällaisessa tilanteessa kaikupulssin teho alittaa selvästi ilmaisukynnyksen, jolloin tutka ei havaitse kohdetta. Tapauksessa 3 joko tutkan suunnittelussa on epäonnistuttu, tai jokin ulkoinen tekijä poikkeaa huomattavasti normaalista. Tällainen tilanne saattaa esiintyä esimerkiksi voimakkaan kanavoitumisen seurauksena, jolloin signaalin vaimeneminen ilmakehässä on huomattavasti normaalia pienempää. Kohteesta saatu kaikupulssi ylittää ilmaisukynnyksen. Koska kulkuajan mittaus on käynnistynyt viimeisestä lähetyspulssista, tutka mittaa kulkuajaksi ajan t_2' . Todellinen kulku-aika on kuitenkin t_2 . Kyseessä on siis jaksontakainen kaiku. Tutka laskee kohteen etäisyyden väärin, koska se käyttää laskennassa aikaa t_2' .

Pulssitutkan teoreettinen *etäisyyden erottelukyky* voidaan laskea samasta kaavasta kuin minimimittausetäisyys. Mikäli kahden kohteen keskinäinen, tutkan säteen suuntainen etäisyysero on pienempi, sulautuvat kohteet yhdeksi kaikusignaaliksi. Etäisyyden erottelukykä merkitään suurella δ_R . (Olli ja Lehto, 1999, ss. 41–42; Richards *ym.*, 2008, s. 783)

$$\delta_R = \frac{\tau \cdot c}{2} \quad (4)$$

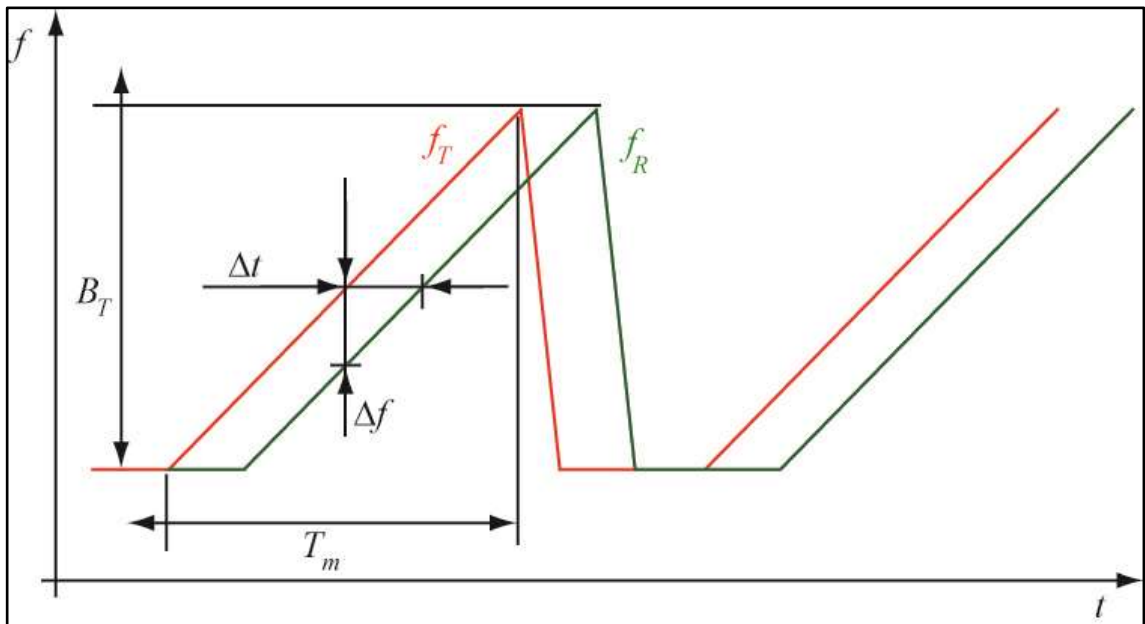
Kulmaerottelukyky riippuu antennin keilanleveydestä. Jos kohteet ovat tutkan sädetä vastaan kohtisuoralla linjalla vähintään keilanleveyden päässä toisistaan, voidaan ne erottaa kahdeksi eri kohteeksi. Keilanleveys määritetään yleensä keilan 3 dB:n pisteiden mukaan. Kulmaerottelukyky, δ_A , voidaan laskea kaavasta

$$\delta_A = \theta_{3dB}[\text{rad}] \cdot R = \frac{\pi}{180^\circ} \theta_{3dB}[\text{°}] \cdot R \quad (5)$$

missä θ_{3dB} on keilanleveys radiaaneina [rad] tai asteina [°]. Käytännössä todellinen erotelukyky on n. 1 – 1,5 kertainen kaavasta saatuun arvoon nähden. (Olli ja Lehto, 1999, s. 42)

2.4 Taajuusmoduloidun kantoaaltotutkan ominaisuuksia

Kantoaaltotutka mittaa ”luonnostaan” kohteen liikkumisen aiheuttaman dopplertaajuuden. Kuten kohdassa 2.2 on todettu, etäisyyden mittaamista varten tarvitaan jatkuvaan läheteeseen jonkinlainen ajastinmerkki. Ajastinmerkkinä voidaan käyttää taajuusmodulaatiota. (Olli ja Lehto, 1999, s. 24)



Kuva 5. FMCW-tutkan lähetys- ja vastaanottosignaalien modulaatio. Perustuu lähteeseen (Olli ja Lehto, 1999).

Taajuusmoduloinut kantoaaltotutka (FMCW-tutka) lähettää vakioamplitudista, taajuudeltaan muuttuvaa signaalia. Moduloivana aaltomuotona voidaan käyttää useita eri aaltomuotoja. Tutkitaan tilannetta jossa moduloiva aaltomuoto on saha-aalto. Tällöin lähetysignaalin taajuus muuttuu lineaarisesti, kunnes moduloivan signaalin jakso alkaa uudelleen. (Olli ja Lehto, 1999, s. 25)

Kuvassa 5 on esitetty lähetysignaali punaisella ja vastaanottosignaali vihreällä. Kun lähetteen pyyhkäisy aika, T_m , pidetään vakiona ja tunnetaan lähetystaajuuden kokonaismuutos, B_T , voidaan kohteen etäisyys laskea mitatun taajuuseron, Δf , perusteella kaavasta (Olli ja Lehto, 1999, s. 25)

$$R = \frac{c \cdot \Delta f}{2 \cdot B_T} \cdot T_m \quad (6)$$

Kaavassa 6 on oletettu, että kohde on paikallaan, jolloin dopplertaajuutta ei synny. Maalin liikkuessaa pitää dopplertaajuuden vaikutus ottaa huomioon. Mikäli moduloiva signaali on kolmioaalto, voidaan dopplertaajuuden vaikutus huomioda helposti, koska taajuuden kasvaessa on sen vaikutus $\Delta f - f_D$ ja vastaavasti taajuuden laskiessa $\Delta f + f_D$. Oikea Δf saadaan siis keskiarvon perusteella. (Olli ja Lehto, 1999, s. 25)

2.5 Robinin 3D-tutka

Tutka koostuu kahdesta erillisestä tutkasta, horisontaali- ja vertikaalitutkasta, sekä niiden tietoja käsittelevästä palvelimesta.

Horisontaalitutka

Horisontaalitutka on S-taajuusalueen pulssitutka. Kohdan 2.2 mukaisesti se on myös:

- yksipaikkatutka, koska sen lähetin ja vastaanotin ovat samassa paikassa ja se käyttää samaa antennia sekä lähetykseen että vastaanottoon
- ensiötutka, koska kaikupulssi syntyy kohteesta heijastuneesta lähetyssignaalista.

Tutkan antennin keilanleveydeksi on ilmoitettu 1.8 astetta ja pulssinpituudeksi 80 ns. 1,8 asteen keilanleveydellä saavutetaan kaavan 5 perusteella 1500 m etäisyydellä kulmaerottelukyky:

$$\delta_A = \frac{\pi}{180^\circ} \theta_{3dB} [^\circ] \cdot R = \frac{\pi}{180^\circ} 1,8^\circ \cdot 1500 \text{ m} \approx 47 \text{ m}.$$

80 ns pulssinpituus vastaa kaavan 4 perusteella etäisyserottelukykä:

$$\delta_R = \frac{\tau \cdot c}{2} = \frac{80 \cdot 10^{-9} \text{ s} \cdot 3 \cdot 10^8 \text{ m/s}}{2} = 12 \text{ m}.$$

Vertikaalitutka

Vertikaalitutka on X-taajuusalueen taajuusmoduloitu kantoaalto (FMCW). Kohdan 2.2 mukaisesti se on myös:

- kaksipaikkatutka, koska se käyttää eri antennia lähetykseen ja vastaanottoon
- ensiötutka, koska kaikupulssi syntyy kohteesta heijastuneesta lähetyssignaalista.

Tutkan keilanleveydeksi on ilmoitettu 0.95 astetta, jolla saavutetaan 1500 m etäisyydellä kulmaerottelukyky:

$$\delta_A = \frac{\pi}{180^\circ} \theta_{3dB} [^\circ] \cdot R = \frac{\pi}{180^\circ} 0,95^\circ \cdot 1500 \text{ m} \approx 25 \text{ m}.$$

Tarkempia tietoja tutkien ominaisuuksista on esitetty liitteessä G.

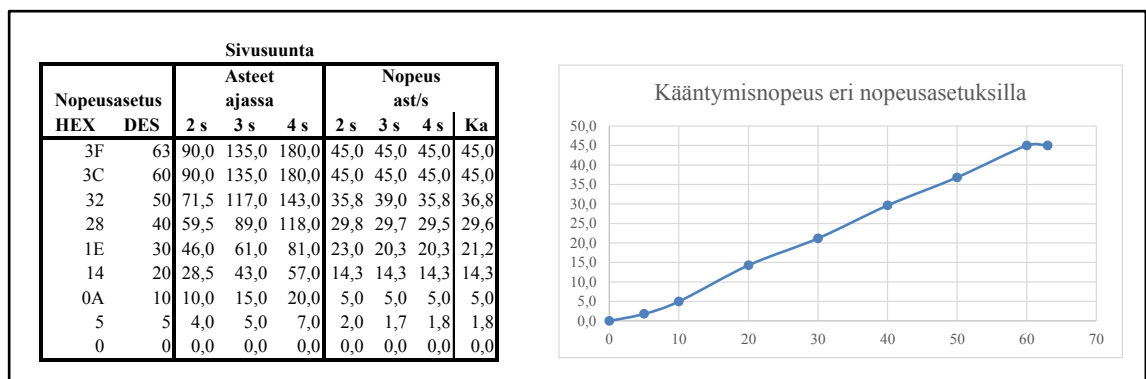
3. KÄÄNTÖPÄÄN PARAMETRIEN SELVITYS

Diplomityön teettäjän esityksestä kääntöpäänä käytetään tässä diplomityössä 2B Security Systemsin valmistamaa PT-1020 kääntöpäätä. Sen eri kokoonpanovaihtoehdoista valittiin PT-1020PRS, jossa hyötykuorma asennetaan kääntöpään sivulle. PRS-mallissa Pelco ”D” lähetin-vastaanotin on sisäänrakennettu ja kääntöpää kykenee käsittelemään myös massaltaan suurempaa hyötykuormaa. Sisäänrakennetun lähetin-vastaanottimen vuoksi oli valittavana vain 24VDC käyttöjännite. Tämä aiheuttaa erillisen teholähteen tarpeen, mutta on toisaalta käyttäjälle turvallisempi. 24 VDC-malli mahdollistaa myös suuremmat kääntönopeudet sekä korkeus- että sivusuunnassa kuin päälle asennettavassa mallissa. Suuremman hyötykuorman käsittelyn vuoksi korkeussuunnan ala- ja yläraja-arvot ovat pienemmät kuin päälle asennettavassa mallissa, mutta maksimi-arvot todettiin riittäviksi suhteessa kohteen korkeuteen ja etäisyyteen. Kääntöpään ominaisuuksia on esitelty liitteessä D.

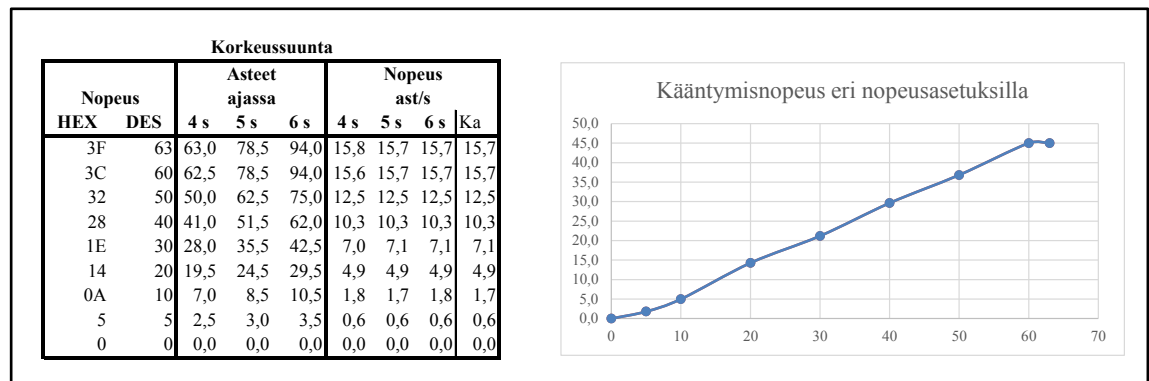
3.1 Kääntöpään kääntymisnopeus

Kääntymiskulman suuruus määräytyy nopeuden ja kääntymisajan perusteella. Tästä syystä ohjelmiston on kyettävä laskemaan oikea kääntymisaika ja pysäyttämään kääntöpää, kun tämä aika on kulunut.

Koska kääntymisnopeus Pelco ”D” -protokollassa on paljas luku, on selvitettävä, mikä on todellinen kääntymisnopeus milläkin lukuarvolla. Tähän käytettiin kohdassa 7.1 esitettyä ohjelmaa. Kuviin 6 ja 7 on koottu testaamalla saatuja kääntymisnopeusarvoja eri nopeusasetuksilla ja kääntöajoilla. Mittausten perusteella laskettu keskimääräinen kääntymisnopeus on esitetty kuvaajana.



Kuva 6. Sivusuunnan kääntymisnopeuden mittausarvot ja kuvaaja.



Kuva 7. Korkeussuunnan kääntymisnopeuden mittausravot ja kuvaaja.

3.2 Kääntöpään asettumistarkkuus

Myös kääntöpään asettumistarkkuutta pyrittiin selvittämään erilaisilla testeillä. Mittaus-tulokset on esitetty liitteessä D. Seuraavassa käydään läpi käytetyt testausmenetelmät ja niistä saadut tulokset.

Sivusuunnan asettumistarkkuus

Mittausjärjestelyssä ohjattiin kääntöpään sivusuuntaa seuraavien sekvenssien mukaisesti (H = kotiasema, R = oikealle, L = vasemmalle):

1. H → R, 2 s → L, 4 s → R. 4 s → L, 4 s → R. 4 s → L, 4 s → R. 4 s → L, 4 s → R. 4 s → L, 4 s → R. 4 s → L, 4 s → R. 4 s → H
2. H → R, 2 s → H → R, 2 s → H → L, 2 s → H → L, 2 s, → H → R, 2 s → H
3. H → R, 1 s → R, 1 s → R, 1 s → H → R, 3 s → H → L, 1 s → L, 1 s → L, 1 s → H → L, 3 s → H.

Ensimmäisessä testissä tutkittiin, kuinka hyvin kääntöpää asettuu aina samaan suuntaan, kun sitä ajetaan kahden raja-arvon välillä edestakaisin.

Toisessa testissä tutkittiin, miten kääntöpää asettuu, kun sitä ajetaan vasemmalle ja oikealle siten, että aina lähdetään liikkeelle kotiasemasta. Tämä testi tehtiin myös noin 6,1 kg painavan, kameran painojakaumaa jäljittelevän kappaleen kanssa, jolla pyrittiin simuloimaan todellista käyttötilannetta.

Kolmannessa testissä tutkittiin, mikä on lopullisen suunnan ero, kun kyseiseen suuntaan ajetaan ensin lyhyillä toistuvilla ajoilla ja sitten yhdellä, lyhyiden ajojen yhteisaikaa vastaavalla ajolla.

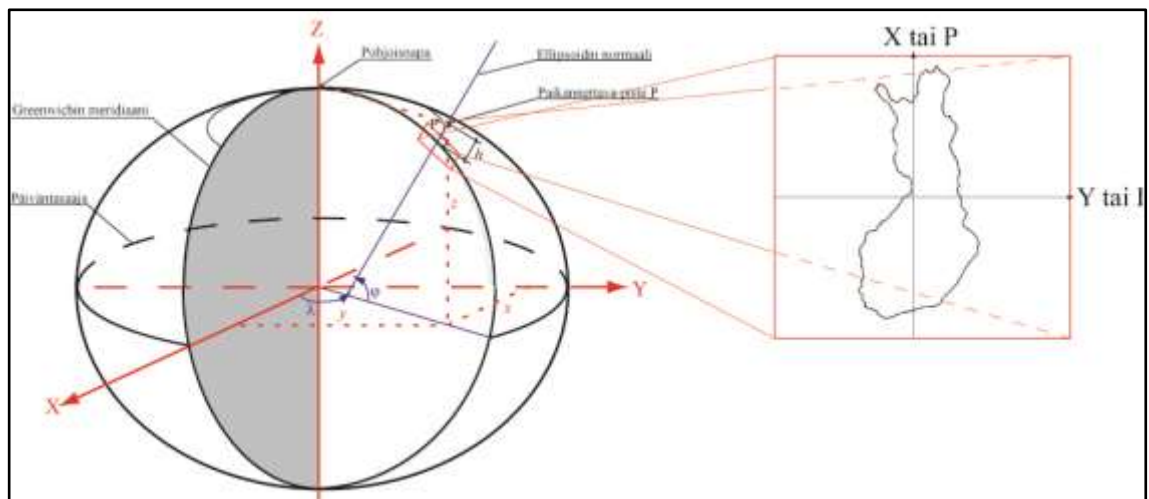
Ensimmäisessä testissä havaittiin, että kääntöpään molemmat ääripisteet siirtyvät jokaisella ajolla noin 4° vasemmalle.

Kameran massalla ei ole vaikutusta suuntauksen tarkkuuteen, mikäli kääntöpään ja kameran kiinnitykset ovat riittävän tukevat. Saattaa kuitenkin olla tarpeen ohjelmallisesti pehmentää kääntöpään liikkeelle lähtöä ja pysäytystä.

4. KOORDINAATISTOT JA KOORDINAATTIJÄRJESTELMÄT

4.1 Koordinaatisto

3D-tutka antaa maalitiedon WGS84-koordinaattijärjestelmässä pituus- ja leveyskulmina sekä korkeutena. Saadun maalitiedon perusteella määritetään suunta- ja etäisyystiedot kamerasta maaliin. Suunta- ja etäisyystiedon laskemiseksi muunnetaan WGS84-koordinaatit tarvittavaan koordinaatistoon sekä maalin korkeus samaan korkeusjärjestelmään kameran korkeuden kanssa.



Kuva 8. Kuvassa on esitetty ellipsoidikeskeinen kolmiulotteinen suorakulmainen koordinaatisto (x -, y - ja z -koordinaatit) ja maantieteellinen koordinaatisto (kulmat λ ja φ sekä korkeus h) sekä karttakoordinaatisto (X/P - ja Y/I -koordinaatit). Perustuu lähteeseen (Laurila, 2012).

Koordinaatistossa voidaan esittää pisteen sijainti täsmällisesti. Paikantamisessa käytetään kolmenlaisia koordinaatistoja, joita havainnollistetaan kuvassa 8. *Ellipsoidikeskeinen kolmiulotteinen suorakulmainen koordinaatisto* määrittää pisteen sijainnin x -, y -, ja z -koordinaattien avulla. *Maantieteellinen koordinaatisto* on kaksiulotteinen kulmamittoja käyttävä koordinaatisto. *Kaksiulotteinen suorakulmainen koordinaatisto* (karttakoordinaatisto) määrittelee pisteen sijainnin X/P - ja Y/I -koordinaattien avulla. Kaksiulotteisissa koordinaatistoissa kolmas ulottuvuus ilmoitetaan korkeutena. (Laurila, 2012, s. 139).

Jotta koordinaatistoa voidaan käyttää, on se realisoitava. Tämä tarkoittaa koordinaatiston kiinnittämistä maapalloon paikallisesti tai maailmanlaajuisesti, jolloin saadaan *koordinaattijärjestelmä*. Koordinaatiston kiinnittää maapalloon *geodeettinen datum* eli vertai-

lujärjestelmä. Vertailujärjestelmä sisältää horisontaalisen- ja vertikaalisen vertailujärjestelmän. Horisontaalinen vertailujärjestelmä kiinnittää koordinaatiston ja vertikaalinen korkeustason (Laurila, 2012, s. 139,154).

Tässä yhteydessä ei tarkastella tarkemmin vertausjärjestelmiä vaan toteamme, että WGS84-koordinaattijärjestelmän vertausjärjestelmä perustuu WGS84-ellipsoidiin, jonka perustana on GRS80-vertausjärjestelmä (Geodetic Reference System 1980) (Hooijberg, 2008, s. 20).

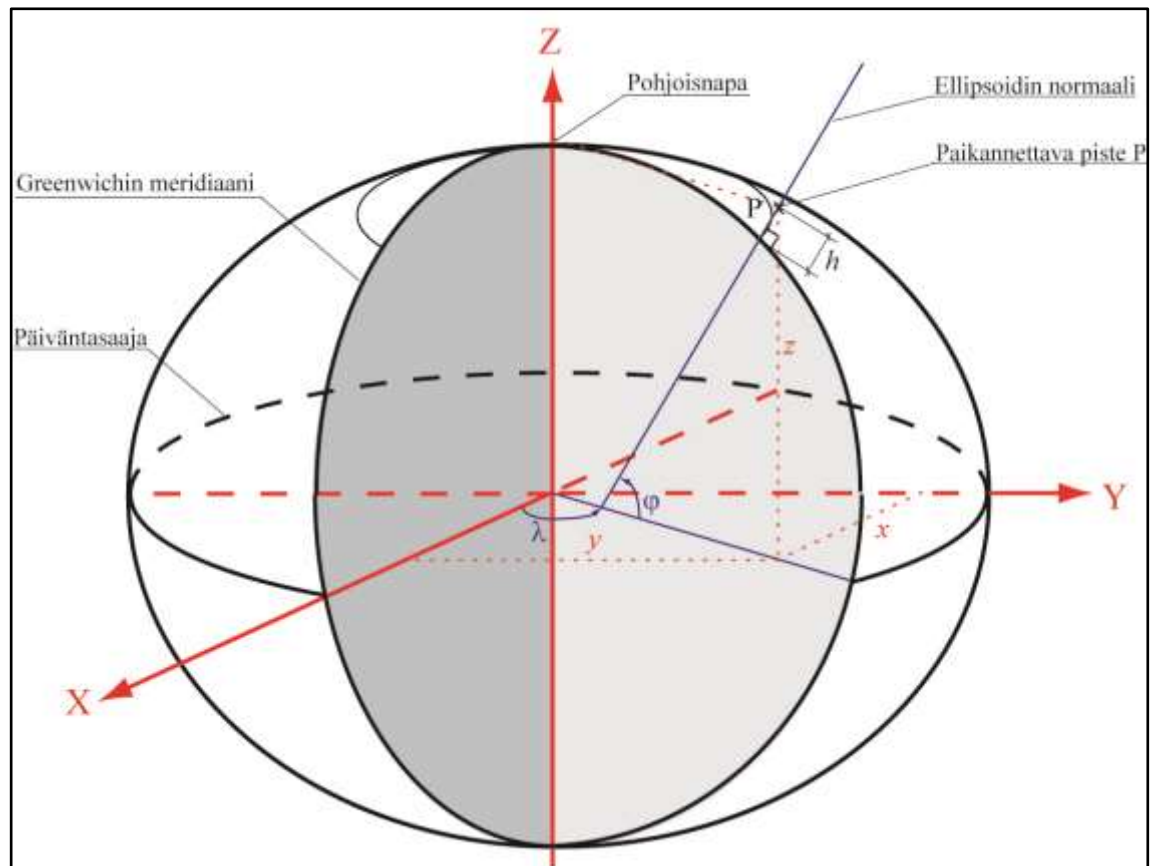
4.2 WGS84-koordinaattijärjestelmä

WGS84-koordinaattijärjestelmä on globaali ja geosentrinen järjestelmä. Järjestelmän suorakulmaisen ellipsoidikeskeisen koordinaatiston realisaation origo on maan massakeskipisteessä (painopisteessä), ja sille on määritelty kolme akselia: X, Y ja Z. Z-akseli on maan pyörimisakseli. X- ja Y-akselit ovat päiväntasaajan tasossa siten, että X-akseli on Greenwichin nollameridiaanitason ja päiväntasaajatason leikkauksessa ja Y-akseli on päiväntasaajatason kohtisuorassa X-akselia vastaan (Hooijberg, 2008, s. 21; Laurila, 2012, s. 160).

WGS84-koordinaattijärjestelmässä käytetään sekä suorakulmaisia, ellipsoidikeskeisiä koordinaatteja että maantieteellisiä koordinaatteja. Edellä mainittujen koordinaattien lisäksi voidaan käyttää myös projektiotasolla määritettyjä suorakulmaisia karttakoordinaatteja, joita käsitellään tarkemmin kohdassa 4.3. (Laurila, 2012, s. 160).

WGS84:n suorakulmaisen ellipsoidikeskeisen koordinaatiston Z-akselin arvot ovat positiivisia maapallon pohjoisella pallonpuoliskolla ja negatiivisia eteläisellä pallonpuoliskolla. Y-akselin arvot ovat positiivisia origon itäpuolella ja negatiivisia origon länsipuolella. X-akselin arvot ovat positiivisia Greenwichin puoleisella pallonpuoliskolla ja negatiivisia vastakkaisella puoliskolla (Laurila, 2012, s. 142).

Maantieteellisessä WGS84-koordinaatistossa paikka ilmoitetaan latitudi (lat , φ)- ja longitudi (lon , λ)-kulmina sekä korkeutena (h) vertausellipsoidin pinnasta (katso kohta 4.4). Latitudista käytetään suomenkielistä nimitystä leveyskulma ja longitudista vastaavasti nimitystä pituuskulma. Leveyskulma mitataan ellipsoidin pinnan normaalin ja päiväntasaajan tason välisenä kulmana päiväntasaajasta pohjoiseen tai etelään. Pituuskulma mitataan kulmana nollameridiaanin tason ja meridiaanin tason välisenä kulmana, nollameridiaanista itään tai länteen (Laurila, 2012, s. 141). Maantieteellisten- ja suorakulmaisten koordinaattien käyttöä on havainnollistettu kuvassa 9.



Kuva 9. Maantieteellisten (λ , φ) sekä suorakulmaisten (x , y , z) koordinaattien vastaavuus. Perustuu lähteeseen (Laurila, 2012, s. 143).

Leveyskulma ja pituuskulma ilmoitetaan yleensä asteina, minuutteina, sekunteina ja sekunnin osina, mutta ne voidaan periaatteessa ilmoittaa minä tahansa kulmayksikköinä. Pallonpuolisko ilmoitetaan yleensä englannin- tai suomenkielen etuliitteillä N tai P (north, pohjoinen), S tai E (south, eteläinen), E tai I (east, itäinen), W tai L (west, läntinen). Laskettaessa voidaan käyttää S- ja E-arvoista negatiivista merkintää ja vastaavasti N- ja W-arvoista positiivista. (Laurila, 2012, s. 141)

Jatkossa tässä esityksessä käytetään maantieteellisten koordinaattien yhteydessä englanninkielisiä lyhenteitä N, S, E ja W.

4.3 EUREF-FIN ja ETRS89-koordinaattijärjestelmät

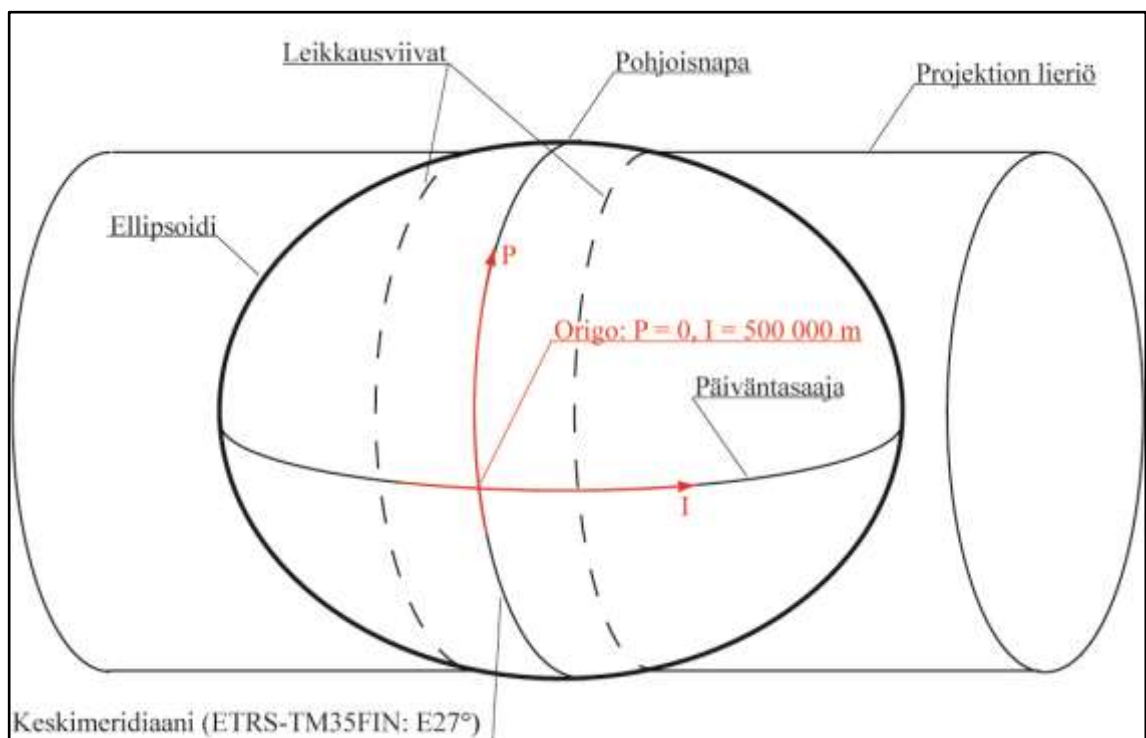
Suomessa käytettävä EUREF-FIN (European Reference Frame – Finland) maailmanlaajuinen karttakoordinaattijärjestelmä perustuu ETRS89-järjestelmään (European Terrestrial Reference System 1989). ETRS89 on kiinnitetty Euraasian mannerlaatan muuttumattomaan osaan, jolloin järjestelmään kuuluvien pisteiden koordinaatit eivät muutu ajan kuluessa. Näin ETRS89 muodostaa Euroopan kattavan yhtenäisen koordinaattijärjestelmän. (Laurila, 2012, s. 161)

ETRS89 on yhtenevä ITRS-järjestelmän (International Terrestrial Reference System) kanssa epokissa 1989.0. Järjestelmien koordinaatistot ovat siis olleet yhtenevät määrittelyhetkellä vuonna 1989 ja sen jälkeen niiden välinen ero muuttuu ajan kuluessa. (Laurila, 2012, s. 161)

Suomessa ETRS89-järjestelmässä toimittaessa, suositellaan käytettäväksi UTM-projektioon perustuvaa suorakulmaista koordinaatistoa. UTM-projektio on leikkaava, poikittaisasentoinen ja oikeakulmainen lieriöprojektiio. (Laurila, 2012, s. 161)

UTM-projektiossa maapallon pinta on jaettu 60 projektiokaistaan, jotka ovat kuuden asteen levyisiä ja kulkevat pohjois-etelä -suunnassa. Suomi sijoittuu kaistoille 34-36. Suomessa on tehty kansallinen poikkeus, jossa koko maa kuvataan kaistan 35 koordinaatistossa keskimeridiaanina 27° itäistä leveyttä (E27°). Tästä kansallisesta projektiosta käytetään nimitystä ETRS-TM35FIN (European Terrestrial Reference System – Transverse Mercator 35 – Finland). (Laurila, 2012, ss. 162–163)

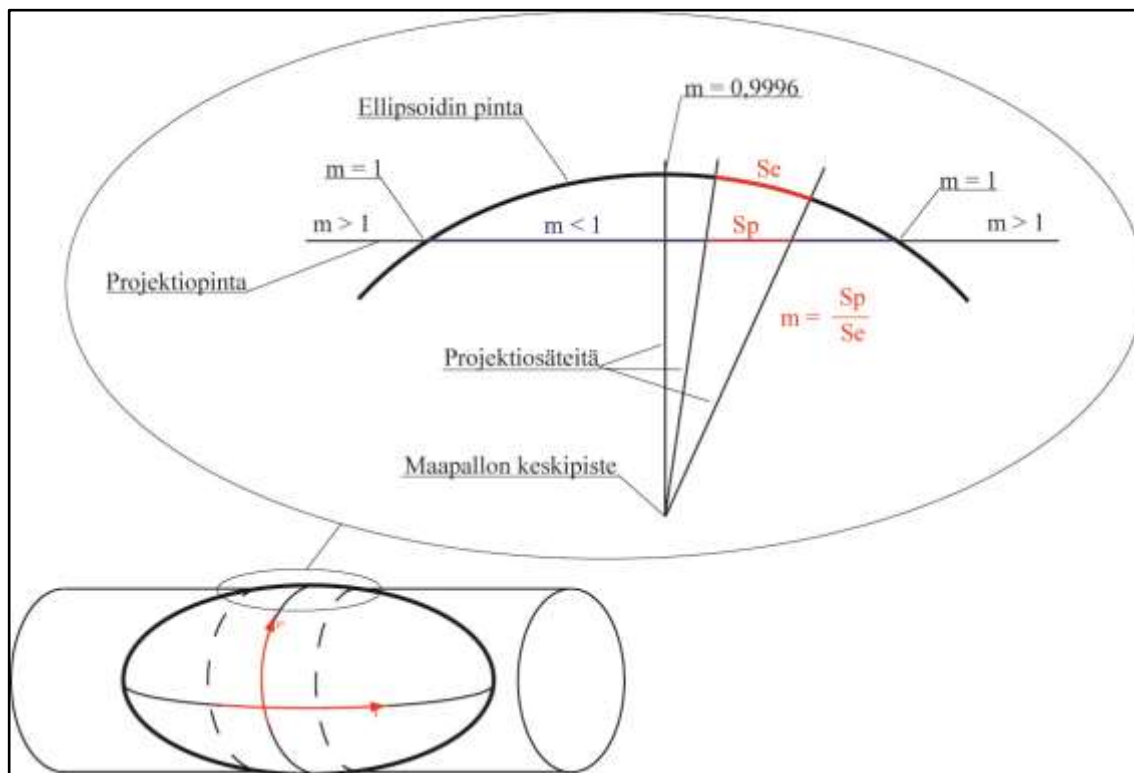
Kuvassa 10 on havainnollistettu ETRS-TM35FIN projektiokaistan muodostumista.



Kuva 10. UTM-projektion periaate sekä tasokoordinaatiston sijoittuminen muodostuneeseen projektiokaistaan. Lieriö voidaan oikaista tasoksi leikkaamalla se pituussuuntaan auki. Perustuu lähteeseen (Laurila, 2012, s. 146,151).

UTM-projektion suorakulmaisissa koordinaateissa käytetään pohjoiskoordinaattia (N, Northing), joka ilmoittaa etäisyyden metreinä päiväntasaajalta projektion pintaa pitkin mitattuna. Itäkoordinaatin (E, Easting) arvo keskimeridiaanilla on 500 000 m. (Laurila, 2012, s. 162)

Jatkossa tässä esityksessä käytetään UTM-projektion koordinaattien yhteydessä suomenkielisiä lyhenteitä P ja I, niiden erottamiseksi WGS84-koordinaattien lyhenteistä.



Kuva 11. Mittakaavakerroin määräytyy projisoituneen matkan ja ellipsoidin pinnalla kuljetun matkan suhteesta. Perustuu lähteeseen (Laurila, 2012, s. 148,151).

Ominaista leikkaavalle projektiolle on, että sen mittakaavakerroin $m = 1$ kahdella leikkausviivalla, $m < 1$ näiden leikkausviivojen välissä ja $m > 1$ leikkausviivojen ulkopuolella. UTM-projektiossa keskimeridiaanilla $m = 0,9996$. Kuva 11. (Laurila, 2012, s. 161)

4.4 Korkeusjärjestelmä

Maapallon pinta on hyvin monimutkainen ja sen käyttäminen korkeuden referenssipinnana on käytännössä mahdotonta.

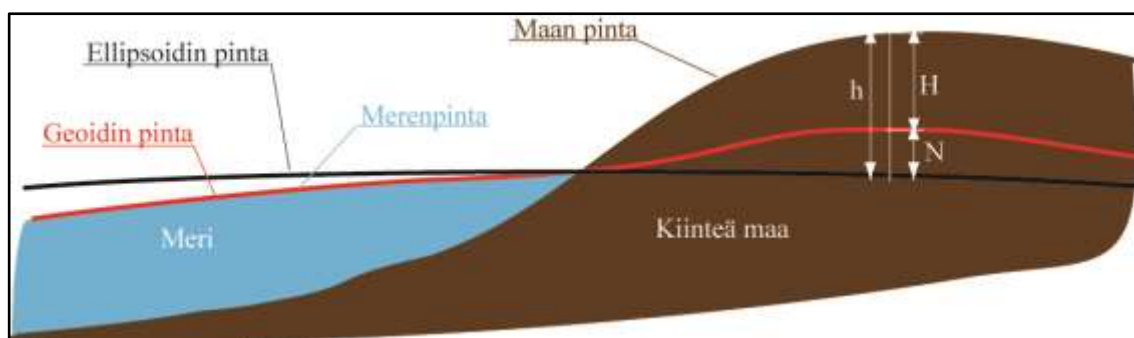
Todellisesta pinnasta on tehty geoidimalli, joka kuvaa maapallon muotoa painovoimakentän mukaisesti. Geoidin pinta noudattaa maan painovoimakentän sama-arvopintaa. Geoidimallin sama-arvopinnaksi on valittu pinta, johon vapaa vedenpinta asettuisi, jos se peittäisi koko maapallon. *Geoidin pinta siis yhtyy valtamerien keskimerenpintaan.* (Laurila, 2012, s. 136)

Geoidin pintaa ei kyetä kuvaamaan matemaattisesti. Tämän vuoksi referenssipinnaksi on valittu pyörähdysellipsoidi. WGS84-koordinaattijärjestelmän vertausellipsoidi on globaali ja geosentrinen malli maapallon pinnasta. (Hooijberg, 2008, s. 127)

Korkeus mitattuna vertausellipsoidin pinnasta on ellipsoidinen korkeus (h), ortometrinen korkeus (H) mitataan geoidin pinnasta ja geoidin korkeus (N) kertoo geoidin pinnan korkeuden vertausellipsoidin pinnasta. (Laurila, 2012, s. 166)

Riippuvuus edellä mainittujen suureiden välillä voidaan siis esittää matemaattisesti kaavalla:

$$h = H + N. \quad (7)$$



Kuva 12. Ellipsoidinen korkeus (h) on ortometrisen korkeuden (H) ja geoidin korkeuden (N) summa. Perustuu lähteeseen (Laurila, 2012, s. 166).

Jotta korkeuksien vertaaminen toisiinsa onnistuu, on niiden oltava samassa korkeusjärjestelmässä. Tässä työssä kaikki laskennassa käytetyt korkeudet esitetään ellipsoidisina korkeuksina ja tarvittaessa muut korkeudet muutetaan ellipsoidisiksi kaavan (7) avulla.

Geoidin korkeus maapallon eri paikoissa vaihtelee, joten sen korkeus pitää määrittää sijainnin perusteella. Tässä diplomityössä geoidin korkeus määritettiin NUVACO:n ”Geoid Height Calculator” -palvelun avulla (<http://www.unavco.org/software/geodetic-utilities/geoid-height-calculator/geoid-height-calculator.html>). Palvelun perusteella geoidin korkeus tutkan ja kameran käyttöpaikalla diplomityötä tehtäessä (lat: 61.630185°, lon: 21.375570°) on 18,81 m.

4.5 Koordinaatti- ja koordinaattijärjestelmämuunnokset

Tässä kohdassa käsitellään koordinaattimuunnoksien ja koordinaattijärjestelmämuunnoksien eroa.

Kuten aikaisemmin WGS84-koordinaattijärjestelmän yhteydessä todettiin, voidaan koordinaattijärjestelmässä paikka esittää kolmella eri tavalla: kolmiulotteisina suorakulmaisina koordinaatteina (x, y, z), maantieteellisinä koordinaatteina (φ, λ) tai kaksiulotteisen projektiopinnan koordinaatteina (P, I) ja sovitun korkeusjärjestelmän mukaisena korkeutena. Muunnokset koordinaattijärjestelmän sisällä ovat matemaattisia ja tarkkoja. (Laurila, 2012, s. 170)

Koordinaattijärjestelmien väliset muunnokset perustuvat kahden koordinaattijärjestelmän välisiin yhteisiin pisteisiin, joiden välille luodaan muunnoskaavat. Koska yhteiset pisteet

mitataan erikseen kummassakin järjestelmässä, sisältyy niihin mittausvirhe. Mittausvirheestä johtuen koordinaattijärjestelmien väliset muunnokset eivät ole virheettömiä, vaikka matemaattiset muunnoskaavat itsessään olisivatkin virheettömiä. (Laurila, 2012, s. 171)

Tässä diplomityössä ei tarvita koordinaattijärjestelmämuunnoksia, koska kaikki laskenta suoritetaan samassa WGS84-koordinaattijärjestelmässä. Sen sijaan tarvitaan koordinaattimuunnos maantieteellisistä koordinaateista (φ, λ) suorakulmaiseen koordinaatistoon. Koordinaattimuunnoksissa käytetään seuraavia Geodeettisen laitoksen muunnoskaavoja: (Häkli, Puupponen ja Koivula, 2009, s. 72)

$$x = (N_R + h) \cos \varphi \cos \lambda \quad (8)$$

$$y = (N_R + h) \cos \varphi \sin \lambda \quad (9)$$

$$z = (N_R(1 - e^2) + h) \sin \varphi \quad (10)$$

z -koordinaatti voidaan laskea myös kaavasta: (Poutanen, 2000, s. 3)

$$z = \left(\frac{b^2}{a^2} \cdot N_R + h \right) \cdot \sin \varphi \quad (11)$$

Edellä esitetyissä kaavoissa 8 – 11 a on ellipsoidin isoakselin puolikas, b on ellipsoidin pikkuakselin puolikas, h on korkeus vertausellipsoidista, e ellipsoidin ensimmäinen eksentrisyys ja N_R poikittaiskaarevuussäde. Suureet e ja N_R lasketaan kaavoista: (Häkli, Puupponen ja Koivula, 2009, s. 72)

$$e = \sqrt{2f - f^2} \quad (12)$$

$$N_R = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (13)$$

Kaavassa 12 f on ellipsoidin litistyssuhde.

N_R voidaan laskea myös kaavasta (Poutanen, 2000, s. 3):

$$N_R = \frac{a^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}} \quad (14)$$

Litistyssuhde ilmoitetaan yleensä käänteislukuna $1/f$. Vakiot a , b ja $1/f$ ovat WGS84-koordinaattijärjestelmässä (Hooijberg, 2008, s. 20):

$$a = 6378137,0 \text{ m}$$

$$b = 6\,356\,752,31424518 \text{ m}$$

$$\frac{1}{f} = 298,257223563$$

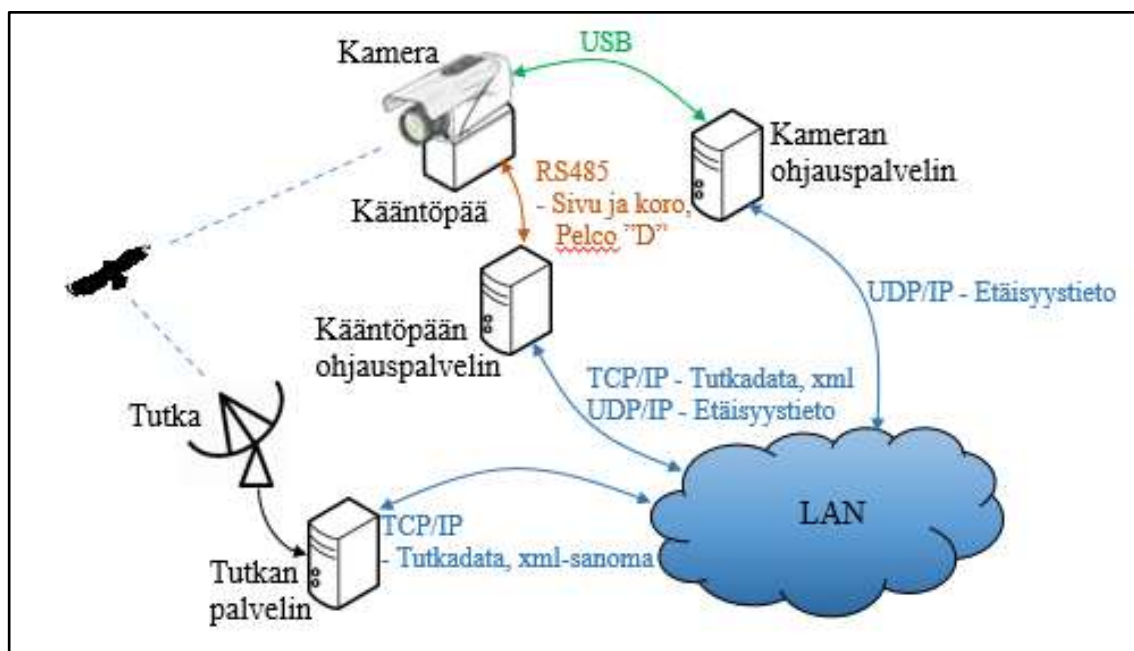
Näiden kolmen vakion suhdetta voidaan kuvata kaavalla: (Laurila, 2012, s. 137)

$$f = \frac{a-b}{a} \quad (15)$$

Yleensä ellipsoidista annetaan vakiot a ja f , jolloin b voidaan laskea kaavasta 15 johdettulla kaavalla:

$$b = a (1 - f) \quad (16)$$

5. KÄYTETYT TIETOLIIKENNEYHTEYDET JA –PROTOKOLLAT



Kuva 13. Tietoliikenneyhteydet ja niillä siirrettävä informaatio.

Tietoliikenneyhteydellä tieto siirretään sovitulla tavalla. Tässä sopimuksessa määritellään miten tieto siirretään, mitä toimenpiteitä tehdään ennen varsinaisen tiedon siirtoa ja miten tarvittaessa varmistetaan tiedon perille meno ja sen oikeellisuus. Sopimusta siitä, miten tieto sovellusten välillä siirretään, kutsutaan protokollaksi. (Stevens, Fenner ja Rudoff, 2003, s. 3; Kozierok, 2005, s. 64)

Protokollat toimivat usein niin kutsutulla palvelin-asiakas -periaatteella, jossa toinen osapuoli ottaa vastaan pyyntöjä ja lähettää sitten niihin vastauksia sovitulla tavalla. Tällä periaatteella toimii esimerkiksi www-palvelimet ja -selaimet. Selain on asiakas, joka lähettää pyyntöjä palvelimelle ja saa vastauksena esimerkiksi jonkin www-sivun sisällön. (Stevens, Fenner ja Rudoff, 2003, s. 3)

Tutka lähettää maalitiedot TCP/IP-sanomina ja järjestelmäkameraa ohjataan UDP/IP-sanomilla. Kääntöpäätä ohjataan RS-845 -yhteydellä ja ohjauksessa käytetään kääntöpään ohjauksen omaa Pelco "D" -protokollaa.

5.1 Piiri- ja pakettikytkentäiset verkot

Yksi tapa käsitellä verkkojen toteutustapaa on jakaa ne piiri- ja pakettikytkentäisiin verkkoihin. Piirikytkentäisessä verkossa yhteys kahden laitteen välillä on fyysisesti olemassa

joko kokoajan, tai se muodostetaan tarvittaessa. Yhteys on joka tapauksessa olemassa vain näiden kahden laitteen välistä tiedonsiirtoa varten. Tällainen verkko on esimerkiksi vanha lankapuhelinverkko, jossa yhteys kahden laitteen välille luodaan soittamalla vastaanottajan numeroon. Kun yhteys on luotu, jotakin puhelinverkossa tarjolla olevaa reittiä pitkin, voidaan puhuminen aloittaa. Tämä yhteys on olemassa, kunnes se puretaan sulkeamalla puhelin. Kumpikaan osapuoli ei voi olla samaan aikaan yhteydessä muihin. Myös koko yhteyden siirtokapasiteetti on varattu siitä riippumatta puhutaanko vai ei. Kun yhteys puretaan ja avataan uudelleen, se voi fyysisesti muodostua eri reittiä, mutta edelleen yhteys on kiinteästi muodostettu laitteiden välille. Pakettikytkentäisissä verkoissa laitteiden välille ei luoda kiinteää reittiä. Data paloittelaa datapaketeiksi, jotka lähetetään vastaanottajalle. Paketit reititetään vastaanottajalle jokainen erikseen ja ne voivat siis kulkea täysin eri reittejä. Vastaanotossa paketit jälleen yhdistetään ja saadaan alkuperäinen data. (Kozierok, 2005, ss. 66–68)

Pakettikytkentäisessä verkossa samalla fyysisellä yhteydellä voi liikkua monien yhteyksien dataa, jolloin verkon käyttöaste kasvaa. Data voi kulkea useita eri reittejä pitkin, mikä voi aiheuttaa sen, että lähetetyt paketit tulevat perille eri järjestyksessä kuin missä ne lähetettiin. On myös mahdollista, että yksi tai useampi paketti hukkuu matkalla. (Kozierok, 2005, s. 68)

5.2 Protokollapino

On olemassa useita erilaisia protokollapinoja, joista yleiskäyttöisin on ISO:n (International Organization for Standardization) julkaisema OSI-malli (Open Systems Interconnection). Malli kuvaa protokollapinoa yleisellä tasolla.

ISO/OSI-malli on otettu virallisesti käyttöön vuonna 1979 ja malli sisältää seitsemän kerrosta. Kolme alinta kerrosta liittyvät tietoliikenteen reititykseen ja tiedon siirtämiseen fyysisellä yhteydellä (esimerkiksi kaapeli tai kuitu). Tiedon siirtoon verkon yli, sovelluksesta toiseen, tarvitaan kuitenkin kaikkia kerroksia. Eri kerroksilla dataa käsitellään datapaketeina, PDU, (Protocol Data Unit) ja jokainen kerros käsittelee dataa omalla tavallaan. Lähetettäessä jokainen kerros voi lisätä datapakettiin omia tunnisteitaan ylemmältä kerrokselta tulleen datapaketin alkuun (otsikko, header), loppuun (loppumerkki, trailer) tai sekä alkuun että loppuun. Vastaanottopäässä jokainen kerros purkaa omat tunnisteensa ennen datapaketin välittämistä ylemmälle kerrokselle. Tästä luonnollisesti seuraa, että sekä lähetys- että vastaanottopäässä on oltava samat protokollakerrokset. (Zdonik ja Katz, 2014, ss. 5–6)

TCP/IP-malli on jatkumo vuonna 1969 aloitetulle ARPANET-projektille (Advanced Research Projects Agency Network). TCP/IP-malli perustuu tiettyihin, mallia luotaessa jo käytössä olleisiin protokolleihin. Malli on saanut nimensä kahdesta protokollasta, jotka muodostavat sen perustan. ISO/OSI- ja TCP/IP-mallien välillä on siis merkittävä ero:

OSI/ISO-malli lähtee liikkeelle ajatuksesta, että protokollat suunnitellaan siihen perustuen, kun taas TCP/IP-malli on luotu mallintamaan jo olemassa olevia protokollia. TCP/IP-mallissa on ISO/OSI-malliin verrattuna käytössä vain neljä kerrosta. Kerrokset ja niiden vastaavuus ISO/OSI-malliin on esitetty taulukossa 1. (Zdonik ja Katz, 2014, ss. 19–20)

Taulukko 1. ISO/OSI-mallin ja TCP/IP-mallien kerrosten vastaavuus. Perustuu lähteeseen (Zdonik ja Katz, 2014).

ISO/OSI-mallin kerros	TCP/IP-mallin kerros
Sovelluskerros (Application Layer)	Sovelluskerros (Application Layer)
Esityskerros (Presentation Layer)	
Istuntokerros (Session Layer)	
Kuljetuskerros (Transport Layer)	Kuljetuskerros (Transport Layer)
Verkkokerros (Network Layer)	Verkkokerros (Network Layer)
Siirtoyhteyskerros (Data Link Layer)	Peruskerros (Network Access Layer, Network Interface Layer)
Fyysinen kerros (Physical Layer)	

5.3 Yhteydetön- ja yhteydellinen kommunikointi

Yhteydellisen (connection-oriented communication) ja yhteydettömän (connectionless communication) kommunikoinnin ero on, että yhteydellisessä kommunikoinnissa lähettäjän ja vastaanottajan välille luodaan yhteys ennen varsinaisen datasiirron aloittamista, kun taas yhteydettömässä kommunikoinnissa data lähetetään vastaanottajalle ilman mitään alkuvalmisteluja (Kozierok, 2005, s. 69; Zdonik ja Katz, 2014, s. 6).

Piirikytkentäinen verkko on aina yhteydellinen, koska päätelaitteiden välillä on fyysinen, tiedon siirtoon varattu kytkentä. Pakettikytkentäisessä verkossa kahden päätelaitteen välillä fyysistä, tiedon siirtoon varattua yhteyttä taas ei ole, vaan datapaketit voivat käyttää useita eri yhteyksiä ja reittejä. Yhteydellinen kommunikointi muodostaakin pakettiverkossa loogisen kytkennän kahden laitteen välille. Yhteydellisessä kommunikoinnissa on erotettavissa neljä eri vaihetta:

- yhteyden muodostus, jolloin yhteys kahden pisteen välille avataan
- neuvottelu, jolloin sovitaan tietyistä yhteyden parametreista
- tiedonsiirto ja sen hallinnointi
- yhteyden purkaminen. (Kozierok, 2005, s. 69)

5.4 TCP/ IP-protokollaperhe

TCP-, UDP- ja IP-protokollat kuuluvat TCP/IP-protokollaperheeseen, johon kuuluu myös useita muita protokollia. Protokollaperhe on nimetty kahden protokollan mukaan, TCP ja IP, koska ne muodostavat protokollaperheen perustan. (Kozierok, 2005, s. 197)

Taulukossa 2 on esitetty muutamia TCP/IP-perheen protokollia TCP/IP-mallin mukaisesti ryhmiteltyinä.

Taulukko 2. Esimerkkejä TCP/IP-mallin eri kerroksilla toimivista protokollista. Perustuu lähteeseen (Zdonik ja Katz, 2014).

TCP/IP-mallin kerros	Kerroksella käytettäviä protokollia
Sovelluskerros	HTTP, FTP, SMTP
Kuljetuskerros	TCP, UDP
Verkkokerros	IP, ARP, ICMP
Peruskerros	Ethernet, PPP, SLIP, FDDI, WLAN

Sovelluskerroksen tehtävä on toimittaa lähetettävä data sovellukselta kuljetuskerrokselle sekä toimittaa vastaanotettava data kuljetuskerrokselta oikealle sovellukselle. Yleisin käytössä oleva protokolla on HTTP-protokolla (Hyper Text Transfer Protocol). Yleisesti käytössä ovat myös FTP (File Transfer Protocol)- ja SMTP (Simple Mail Transfer Protocol)-protokollat. Erilaisia sovelluskerroksen protokollia on olemassa erittäin paljon. Useimmat niistä ovat yksinkertaisia tekstipohjaisia protokollia, jotka käyttävät ASCII-merkistää (American Standard Code for Information Interchange). (Zdonik ja Katz, 2014, s. 35)

Kuljetuskerroksen tehtävä on mahdollistaa tiedonsiirtoyhteyden muodostaminen lähettäjän ja vastaanottajan välille. TCP- ja UDP-protokollat toimivat tällä kerroksella. TCP-protokolla tarjoaa yhteydellisen yhteyden ja UDP-protokolla yhteydettömän yhteyden. (Zdonik ja Katz, 2014, s. 29).

Verkkokerroksen tehtävä on vastata datapakettien reitityksestä verkon läpi. Reitityksen tehtävänä on löytää datapaketille paras reitti sen päämäärään. Yleisin käytössä oleva verkkokerroksen protokolla on IP-protokolla. Tällä kerroksella toimii myös IP-protokollaa tukevia protokollia kuten ICMP (Internet Control Message Protocol) ja ARP (Address Resolution Protocol). (Zdonik ja Katz, 2014, s. 22)

TCP/IP-pinon peruskerros ei varsinaisesti ole samanlainen kerros kuin muut TCP/IP-pinon kerrokset. TCP/IP-standardi ei määrittele tätä kerrosta yksityiskohtaisesti. Peruskerroksen tehtävä on välittää verkkokerrokselta tulevat datapaketit verkkoon ja käytetyn median (kuitu, kuparikaapeli) kautta vastaanottajalle sekä päinvastoin. TCP/IP-malli ei ota

kantaa siihen minkälainen verkkoteknologia (esimerkiksi Ethernet) tai siirtotie (esimerkiksi kuitu tai langaton yhteys) on käytössä. Ainoa vaatimus on, että IP-paketit kyetään välittämään. (Zdonik ja Katz, 2014, s. 20)

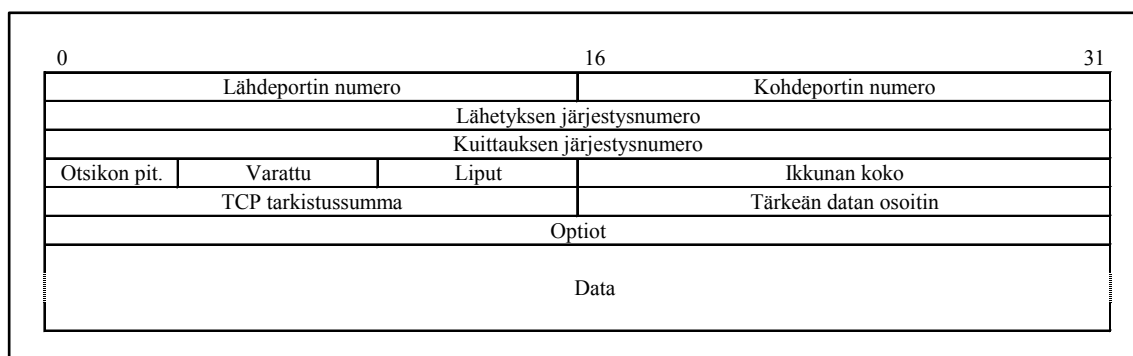
TCP-protokolla

TCP-protokolla tarjoaa yhteydellisen ja luotettavan tavupohjaisen tiedonsiirron. TCP-protokollaa käytettäessä tiedonsiirto tapahtuu kahden laitteen välillä. Yleislähetys (broadcast) ja yhdeltä monelle lähetys (multicast) ei ole mahdollista. (Stevens, 1993, s. 242)

Luotettava tiedonsiirto toteutetaan seuraavien TCP-palvelun ominaisuuksien avulla. Data paloittelaa sopivan kokoisiksi paloiksi, *segmenteiksi*. Kun TCP-segmentti lähetetään, käynnistetään *lähetysajastin*. Jos vastaanottaja ei kuittaa lähetettyä pakettia vastaanotetuksi lähetysajastimeen asetetun ajan kuluessa, paketti *uudelleenlähetetään*. Kun TCP-segmentti vastaanotetaan, lähetetään *vastaanottokuittaus* lähettäjälle. TCP-segmentti sisältää *tarkastussumman* koko segmentistä. Jos tarkastussumma vastaanotossa on virheellinen, paketti hylätään ja vastaanottokuittausta ei lähetetä, jolloin lähettäjä uudelleenlähettää kyseisen segmentin lähetysajastimen ohjaamana. IP-paketit voivat, eriävistä siirto-riteistä johtuen, saapua eri järjestyksessä kuin ne on lähetetty. Tämän vuoksi TCP-palvelu uudelleenjärjestää IP-paketeissa olleet TCP-segmentit ja toimittaa ne oikeassa järjestyksessä vastaanottavalle sovellukselle. Koska IP-paketit voivat myös monistua lähetysten aikana, TCP-palvelu tuhoaa segmentit, jotka ovat monistuneet. TCP-palvelu tarjoaa myös *vuonohjauksen* joka perustuu lähettäjän ja vastaanottajan lähetys- ja vastaanottopuskureihin. Vastaanottaja sallii lähettäjän lähettää vain sellaisen määrän dataa, jonka se pystyy vastaanottamaan puskuriinsa. Tämä estää suurempaan siirtonopeuteen kykenevää laitetta lähettämästä dataa liian nopeasti. (Stevens, 1993, ss. 242–243)

TCP-tiedonsiirto on tavupohjaista tiedonsiirtoa. Tämä tarkoittaa, että TCP ei sisällä mitään tietuumerkkejä, jolloin lähettäjä voi lähettää dataa erikokoisina segmentteinä esimerkiksi 10 tavua, sitten 20 ja sen jälkeen vielä 50 tavua. Vastaanottaja saattaa vastaanottaa tämän lähetysten lukemalla neljä kertaa 20 tavua. TCP-tiedonsiirto ei myöskään ota kantaa siirrettävän datan sisältöön. TCP:n kannalta on yhdentekevää onko data binääristä, ASCII-merkkejä tai jotain muuta. *Tavuvuon tulkinta jää lähettävän ja vastaanottavan sovelluksen tehtäväksi*. (Stevens, 1993, s. 243)

Kuvassa 14 on esitetty TCP-paketin rakenne ja taulukossa 3 on selitetty kenttien sisältö ja merkitys sekä kentän englanninkielinen nimi.



Kuva 14. TCP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 244).

TCP-otsikon koko on 20 tavua, ellei se sisällä valinnaisia ominaisuuksia. Lähde- ja kohdeportin numerot sekä IP-paketin sisältämät lähde- ja kohde IP-osoitteet muodostavat yhdessä *yhteystunnisteen* joka yksilöi yhteyden. Yhteystunnisteesta käytetään toisinaan myös nimitystä soketti (Stevens, 1993, s. 244).

Taulukko 3. TCP-paketin eri kenttien sisältö ja merkitys. Perustuu lähteeseen (Stevens, 1993, s. 244–246, 250, 317).

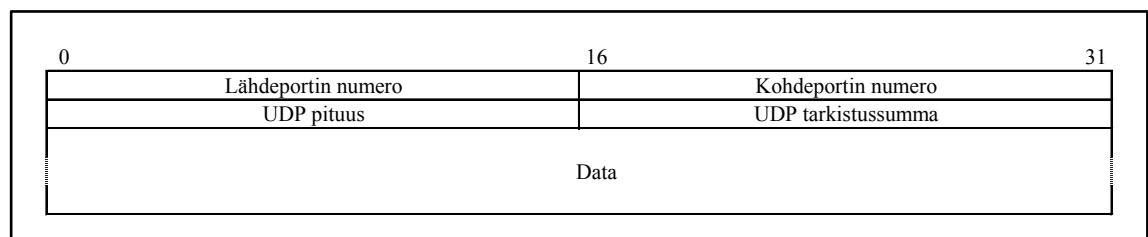
Kenttä	Kuvaus
Lähdeportin numero, 16 bittiä (Source port number)	Yksilöi lähettävän sovelluksen.
Kohdeportin numero, 16 bittiä (Destination port number)	Yksilöi vastaanottavan sovelluksen.
Lähetyksen järjestysnumero, 16 bittiä (Sequence number)	Sisältää segmentissä olevan ensimmäisen datatavun järjestysnumeron koko yhteyden tavuvirrassa. Yhteyttä muodostettaessa tässä kentässä on järjestysnumeron alkuarvo (katso kohta Liput - Synkronoi).
Kuittauksen järjestysnumero, 16 bittiä (Acknowledgement number)	Sisältää sen järjestysnumeron mitä vastaanottaja seuraavaksi odottaa.
Otsikon pituus, 4 bittiä (Header length)	Otsikon pituus 32 bitin sanoina.
Varattu, 6 bittiä (Reserved)	Varattu myöhempään käyttöön.
Liput, 6 bittiä (Flags: <ol style="list-style-type: none"> 1. Tärkeä (URG, Urgent) 2. Kuittaus (ACK, Acknowledgement) 3. Kiire (PSH, Push) 4. Alusta (RST, Reset) 5. Synkronoi (SYN, Synchronize) 6. Loppu (FIN, Finished) 	Kuusi lippua: <ol style="list-style-type: none"> 1. Kertoo, että segmentti sisältää tärkeää dataa ja tärkeän datan osoitin on asetettu. 2. Kertoo, että kuittausnumero on asetettu. 3. Kertoo, että segmentin data on välitettävä sovellukselle mahdollisimman nopeasti. 4. Kertoo, että yhteys on alustettava uudelleen.

	<p>5. Kertoo, että segmentin lähetyksen järjestysnumerokenttä sisältää segmentin lähettäjän ensimmäisen järjestysnumeron.</p> <p>6. Kertoo, että lähettäjä on lopettanut datan lähettämisen ja yhteys voidaan purkaa.</p>
Ikkunan koko, 8 bittiä (Window size)	Kertoo kuinka monta tavua segmentin lähettäjä hyväksyy vastaanotossa.
TCP tarkistussumma, 8 bittiä (TCP checksum)	Koko TCP-segmentistä laskettu tarkistussumma.
Tärkeän datan osoitin, 16 bittiä (Urgent pointer)	Kentän sisällöllä on merkitystä vain, jos tärkeän-lippu on asetettu. Kentän arvo sisältää luvun, joka on lisättävä lähetyksen järjestysnumeron arvoon, jotta saadaan viimeisen tärkeän datan tavun järjestysnumero.
Optiot, vaihteleva määrä bittejä (Options)	Kenttä sisältää valinnaiset tiedot TCP-segmentissä. Esimerkiksi segmentin maksimikoko.
Data, vaihteleva määrä bittejä (Data)	Lähetetty data.

UDP-protokolla

UDP-protokolla on yksinkertainen, sanomapohjainen kuljetuskerroksen protokolla. Jokainen lähetys tuottaa yhden UDP-paketin, joka puolestaan pakataan yhteen IP-pakettiin. *UDP-protokolla ei varmista datan perillemeno.* (Stevens, 1993, s. 163)

Kuvassa 15 on esitetty UDP-paketin rakenne ja taulukossa 4 on selitetty kenttien sisältö ja merkitys sekä kentän englanninkielinen nimi.



Kuva 15. UDP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 164).

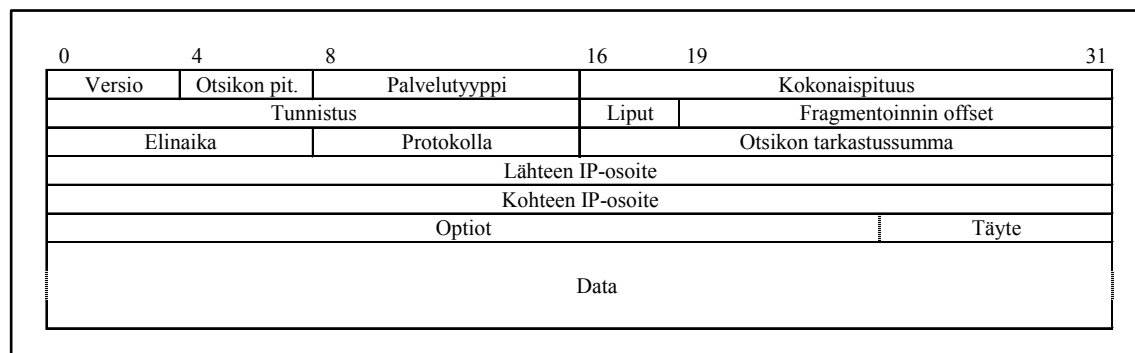
Taulukko 4. UDP-paketin eri kenttien sisältö ja merkitys. Perustuu lähteeseen (Stevens, 1993, s. 244–246,250,317).

Kenttä	Kuvaus
Lähdeportin numero, 16 bittiä (Source port number)	Yksilöi lähettävän sovelluksen.
Kohdeportin numero, 16 bittiä (Destination port number)	Yksilöi vastaanottavan sovelluksen.
UDP pituus, 16 bittiä (UDP length)	UDP-paketin pituus 8 bitin tavuina.
UDP tarkistussumma, 16 bittiä (UDP checksum)	Koko UDP-paketista laskettu tarkistussumma.
Data, vaihteleva määrä bittejä (Data)	Lähetetty data.

IP-protokolla

Verkkokerroksen IP-protokolla on suunniteltu toimimaan pakettikytkentäisissä tietoliikenneverkoissa. Protokollan päätehtävä on toteuttaa yhteydetön datapakettien toimitus päätelaitteelta toiselle. *IP-protokolla ei lupaa luotettavaa pakettien toimitusta*. Tästä seuraa, että haluttaessa luotettava tiedonsiirto, pitää protokollapinon ylempien kerrosten tarjota mekanismit, joilla varmistetaan pakettien luotettava toimitus. (Stevens, 1993, s. 47; Zdonik ja Katz, 2014, s. 23)

Päätelaitteet voivat olla joko samassa verkossa tai eri verkoissa. Eri verkkojen välillä on reitittämiä, jotka huolehtivat datapakettien reitityksestä oikeisiin verkkoihin ja edelleen oikealle päätelaitteelle. Tämä toteutetaan IP-otsikon avulla, jossa on osoite ja ohjaustietoa. IP-protokolla mahdollistaa myös pitkien datapakettien segmentoinnin pienemmiksi paketeiksi (jakamisen ”alipaketteihin”) sekä pakettien uudelleen kokoamisen. Segmentointia ja uudelleen kokoamista tarvitaan, jos paketit liikkuvat verkoissa, joissa suurin sallittu pakettikoko vaihtelee. Jokainen paketti käsitellään itsenäisenä. (Zdonik ja Katz, 2014, s. 23)



Kuva 16. IP-paketin rakenne. Perustuu lähteeseen (Stevens, 1993, s. 48).

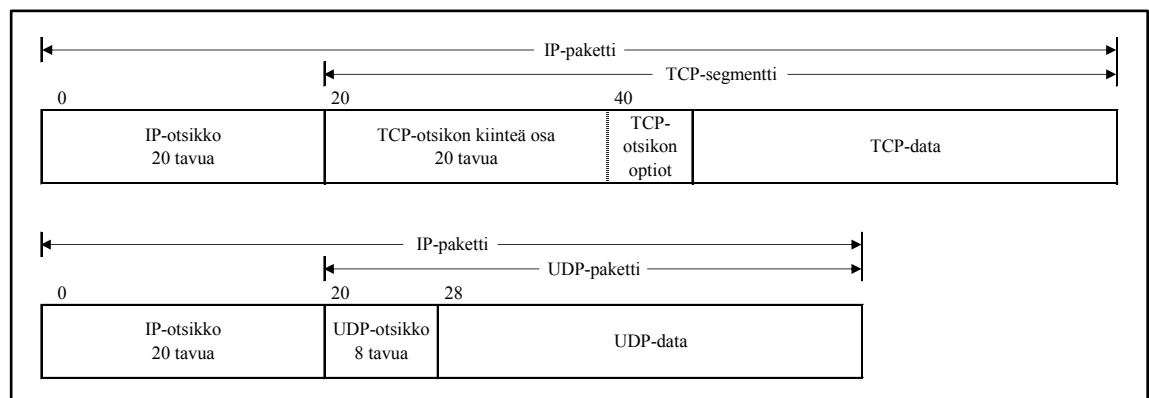
Kuvassa 16 on esitetty IP-paketin rakenne ja taulukossa 5 on selitetty kenttien sisältö ja merkitys.

Taulukko 5. IP-paketin eri kenttien koko ja käyttötarkoitus. Perustuu lähteeseen (Stevens, 1993, s. 25–26, 48–51, 169).

Kenttä	Kuvaus
Versio, 4 bittiä (Version)	Protokollaversio, käytössä versiot 4 ja 6.
Otsikon pituus, 4 bittiä (Header length)	Otsikkokenttien pituus 32 bitin sanoina, optiot mukaan luettuna.
Palvelutyyppi, 8 bittiä (TOS, Type of Service)	3 bittiä etuoikeusasetukselle jota ei nykyisin käytetä. 4 bittiä palvelutypille: minimoi viive, maksimoi läpäisy, maksimoi luotettavuus, minimoi kustannus. 0000 vastaa normaali palvelua. Vain yksi bitti voi olla tilassa 1. Yksi merkityksetön bitti joka pitää aina olla 0.
Kokonaispituus, 16 bittiä (Total length)	IP-paketin (tai alipaketin) kokonaispituus 8 bitin tavuina.
Tunnistus, 16 bittiä (Identifier)	IP-paketin tunnistuskenttä. Yleensä kentän arvo kasvaa yhdellä jokaisessa lähetetyssä IP-paketissa. Jos paketti joudutaan fragmentoimaan, on kaikilla samasta IP-paketista tuotetuilla alipaketeilla sama tunnistuskenttä.
Liput, 3 bittiä 1. Varattu (Reserved) 2. Älä fragmentoi (DF, Don't fragment) 3. Lisää fragmentoituja paketteja (MF, More Fragment)	Kolme lippua: 1. Varattu myöhempään käyttöön. 2. Jos lippu on asetettu, ei IP-pakettia saa fragmentoida. Jos IP-pakettia ei pystytä välittämään se tuhoetaan ja lähetetään virhesanoma paketin lähettäjälle. 3. Asetettu kaikissa muissa alkuperäisen IP-paketin alipaketeissa, paitsi viimeisessä.
Fragmentoinnin offset, 13 bittiä (Fragment offset)	Kun alkuperäinen IP-paketti on fragmentoitu, tämä kenttä kertoo alipaketin datakentän ensimmäisen tavun alkukohdan laskettuna alkuperäisen IP-paketin datakentän alusta lukien.
Elin aika, 8 bittiä (TTL, Time to Live)	Lähetäjä asettaa elinajan. Jokainen reititin vähentää arvoa yhdellä ja jos kenttä saa arvon 0, IP-paketti tuhoetaan ja lähetetään virhesanoma paketin lähettäjälle.
Protokolla, 8 bittiä (Protocol)	Kertoo minkä protokollan dataa IP-paketti sisältää. Esimerkiksi arvo 6 tarkoittaa TCP:tä ja arvo 17 UDP:tä.
Otsikon tarkastussumma, 16 bittiä (Header checksum)	Otsikkokentistä laskettu tarkistussumma.
Lähteen IP-osoite, 32 bittiä (Source IP address)	Lähetäjän IP-osoite.
Kohteen IP-osoite, 32 bittiä	Vastaanottajan IP-osoite.

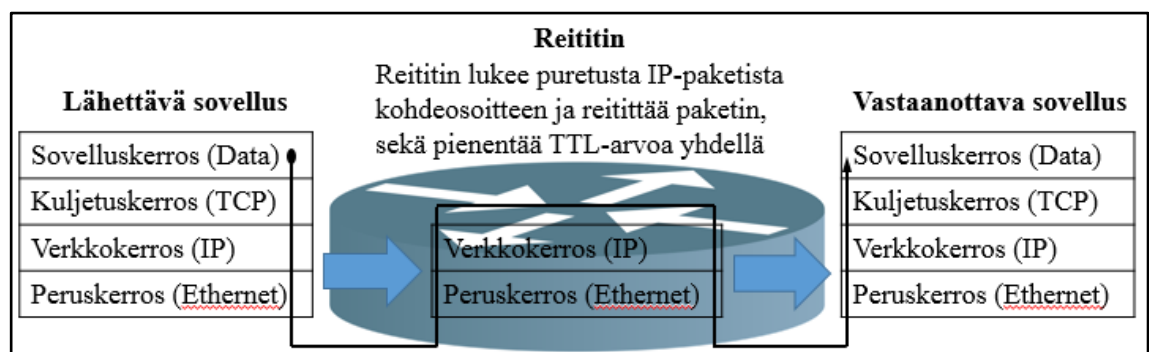
(Destination IP address)	
Optiot, vaihteleva määrä bittijä (Options)	Lista valinnaisista tiedoista IP-paketissa. Esimerkiksi reitin talletus tai lähdereititys asetus.
Täyte, vaihteleva määrä bittijä (Padding)	0-bittijä joilla varmistetaan, että otsikon pituus on 32 bitin kerrannainen.
Data, vaihteleva määrä bittijä (Data)	Lähetetty data.

TCP/IP-pinon kuljetuskerrokselta tuleva paketti (esimerkiksi TCP- tai UDP-paketti) kapseloidaan (eng. encapsulation) IP-paketin sisään. Kuvassa 17 on esitetty tämän kapseloinnin periaate.



Kuva 17. TCP- ja UDP-pakettien kapselointi IP-paketin sisään. perustuu lähteeseen (Stevens, 1993, s. 163,244).

Kuvassa 18 on esitetty eri pakettien muodostuminen ja purkaminen kun data siirretään kuljetuskerrokselta verkkoon ja edelleen reitittimen yli vastaanottajalle.



Kuva 18. Datan siirtyminen lähettävältä sovellukselta vastaanottavalle sovellukselle. Perustuu lähteeseen (Stevens, 1993).

Lähtävä sovellus lähettää datan kuljetuskerrokselle, joka muodostaa TCP-segmentin. Kuljetuskerros siirtää segmentin verkkokerrokselle joka kapseloi segmentin IP-pakettiin. Verkkokerros siirtää IP-paketin peruskerrokselle joka tässä tapauksessa kapseloi paketin

ethernet-kehykseen ja lähettää sen siirtomediaan. Reitin purkaa IP-paketin ethernet-kehuksesta, pienentää IP-paketin TTL-kentän arvoa yhdellä, lukee kohdeosoitteen, kapseloi IP-paketin uudelleen ethernet-kehykseen ja reitittää sen eteenpäin. Vastaanottopäässä kapselointi puretaan ja data toimitetaan sovellukselle. (Stevens, 1993)

5.5 RS-485 liitäntä ja Pelco ”D” protokolla

Kääntöpään ohjaus on toteutettu RS-485 sarjaliikenneväylällä ja kääntöpää saa komentoja Pelco ”D” -protokollan mukaisilla komennolla. RS-485 väylä on RS-422 väylän laajennus. Niillä on samat siirtomatkat ja siirtonopeudet, mutta RS-485 mahdollistaa useamman lähetimen ja vastaanottimen liittymisen samaan väylään. RS-485 mahdollistaa myös ns. ”multidrop”-väylän kaksijohtimisena seuraavilla raja-arvoilla:

- siirtolinjan pituus maksimissaan 1200 m
- siirtonopeus maksimissaan 10 Mbps
- maksimissaan 32 lähetintä samassa väylässä
- maksimissaan 32 vastaanotinta samassa väylässä. (Mackay *ym.*, 2004, s. 53)

Tiedonsiirtolinjan pituus ja maksimisiirtonopeus riippuvat toisistaan. Poikkipinnaltaan 24 AWG (american wire gauge) kierretyllä parikaapelilla ja 1200 m siirtolinjan pituudella maksiminopeus on noin 90 kbps. Vastaavasti maksiminopeudella (10 Mbps) siirtolinjan pituus saa olla maksimissaan noin 6 m. (Mackay *ym.*, 2004, s. 53)

RS-485 väylän lähetin voi olla kolmessa tilassa: looginen 1 tai 0 sekä suuri-impedanssinen tila. Tämä mahdollistaa useiden lähettimien kytkeytymisen samaan väylään kuitenkin niin, että vain yksi voi olla aktiivinen kerrallaan. Väylä ei vaadi terminointia lyhyillä siirtolinjoilla tai pienillä siirtonopeuksilla. Mikäli terminointi tarvitaan, terminointivastusten tulee olla resistanssiltaan samansuuruisia kuin siirtolinjan impedanssi. Kierretyllä parikaapelilla arvo on tyypillisesti 120 Ω . (Mackay *ym.*, 2004, s. 54)

RS-485 väylä on differentiaalinen. Tämä tarkoittaa sitä, että signaalitasoja ei mitata absoluuttisesti maatasoa vastaan, vaan signaalin tila määritellään kahden johtimen, A ja B, välisenä jännite-erona. Jos A-johtimen jännite on pienempi kuin B-johtimen, erojännite on negatiivinen, on kyseessä looginen 1, kun taas päinvastaisessa tilanteessa on kyseessä looginen 0. Päästöaso on ± 200 mV. (Mackay *ym.*, 2004, s. 54)

Koska käytetyssä tietokoneessa ei ole RS-485 porttia, käytetään USB-RS-485 -sovitinta. Sovittimeksi valittiin USB-COMi -sovitin sen edullisuuden ja monipuolisuuden vuoksi. Sovittimen asetukset on esitetty liitteessä A punaisilla korostuksilla.

Pelco ”D” -protokollan kuvaus on esitetty liitteessä B. Seuraavassa on selitetty tässä sovelluksessa käytetyt protokollan ominaisuudet. Tekstissä käytetään heksadesimaalilukujen etuliitteenä merkintää ”0x” ja binäärilukujen etuliitteenä ”0b”. Kuvassa 19 on esitetty protokollan sanomarakenne. Kolmen ensimmäisen tavun sisällöt ovat tässä tapauksessa

vakiot, koska ne eivät sisällä informaatiota, jota olisi tarve muuttaa komentojen muodostuksessa.

Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
Synkronointi	Osoite	Komento 1	Komento 2	Sivunopeus	Koronoisuus	Tarkastussumma
0xFF	0x01	0x00	0b000D ULR 0	0x00...0x40	0x00...0x3F	0x55

Kuva 19. Pelco "D" –protokollan sanomarakenne. Tavun 4 arvo on esitetty binäärisenä, muut tavut ovat heksadesimaalimuodossa.

Tavu 1

Tiedonsiirron synkronointitavu, jonka arvo on aina 0xFF.

Tavu 2

Laiteosoite, joka kääntöpäällä on asetettu kiinteästi arvoon 0x01.

Tavu 3

Komentotavu 1, joka on tarkoitettu lähinnä kameran toimintojen ohjaamiseen. Tässä tapauksessa tavua ei ole tarvetta muuttaa, joten se on asetettu aina arvoon 0x00.

Tavu 4

Komentotavu 2, jonka kolmella ensimmäisellä bitillä ohjataan kameran tarkennusta ja rajausta. Tässä tapauksessa kyseiset bitit on asetettu aina arvoon "0". Biteillä D (Down), U (Up), L (Left), R (Right) määritetään kääntöpään kääntymissuunnat. Kääntöpää pysäytetään nollamalla kyseisen suunnan molemmat bitit (D ja U tai L ja R). Nopeuden asettaminen 0 arvoon ei riitä. Viimeinen bitti on protokollassa määritelty olemaan aina "0".

Tavu 5

Tavu määrittää sivusuuntaisen liikkeen nopeuden. Arvot 0x00... 0x3F ovat normaali nopeuksia ja 0x40 on "turbo" nopeus. Siirtyminen turbo-nopeudesta normaaliin ei ole jouhevaa.

Tavu 6

Tavu määrittää pystysuuntaisen liikkeen nopeuden. Arvot 0x00... 0x3F.

Tavu 7

Tavu sisältää sanoman tarkastussumman, joka lasketaan modulo 256 menetelmällä. Laskentaan otetaan mukaan kaikki muut tavut paitsi synkronointitavu. Modulo 256 summa lasketaan seuraavan esimerkin mukaan.

	10-järjestelmässä	Heksalukuna	Binäärisenä
Tavu 2	10	0A	0000 0001
Tavu 3	136	88	1000 1000
		<i>Summa</i>	<i>0000 0001</i>
Tavu 4	144	90	1001 0000
		<i>Summa</i>	<i>1 0010 0010</i>
Tavu 5	32	20	0010 0000
		<i>Summa</i>	<i>1 0100 0010</i>
Tavu 6	0	0	0000 0000
		<i>Summa</i>	<i>1 0100 0010</i>
Poitetaan ylivuoto	66	42	0100 0010

Kuva 20. Modulo 256 tarkastussumman laskenta binäärisenä.

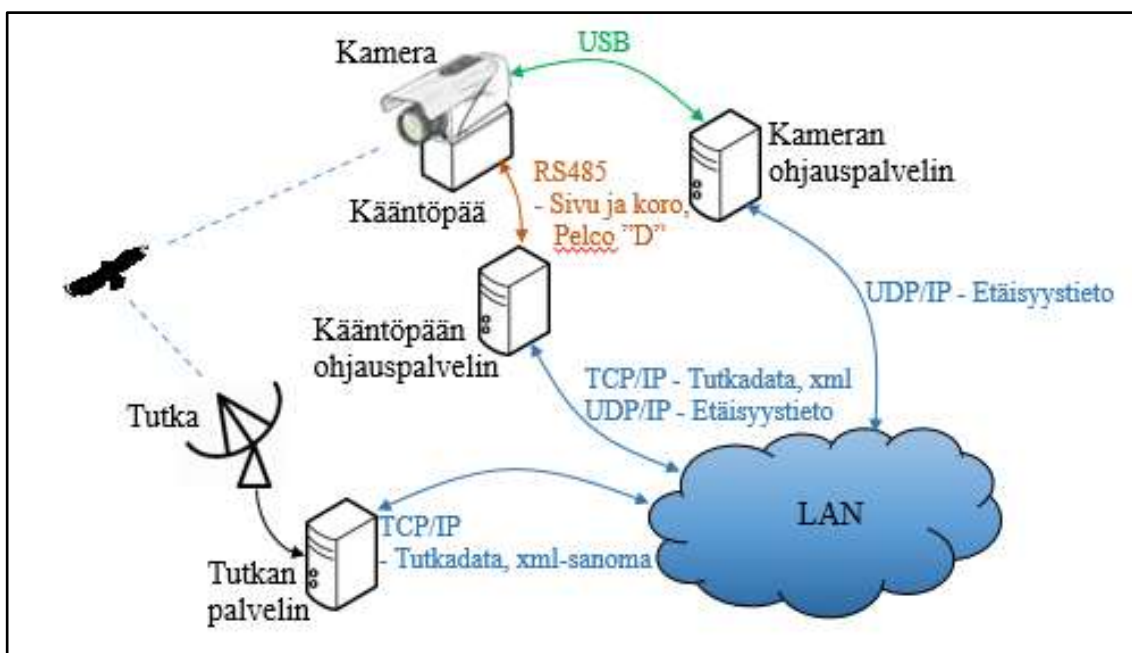
Kuvassa 20 on esitetty tarkastussumman laskenta. Kaikkien summassa mukana olevien tavujen arvot lasketaan yhteen. Koska tavun pituus on rajattu 8 bittiin, ”unohdetaan” ylivuoto ja summasta käytetään vain 8 alinta bittiä. Esimerkissä lopullinen tarkastussumman arvo on 0x42.

Kääntöpään ohjauskomennot muodostetaan ottamalla kuvan 19 perusteella tehty sanomarakko: 0xFF, 0x01, 0x00, 0x00, 0x40, 0x3F, 0x00 ja muuttamalla siitä tarpeen mukaan tavujen 3,4 ja 7 sisältöjä.

Erillisenä komentona tarvitaan edellä mainitusta mallista poikkeava komento, jolla kääntöpää ajetaan kotiasemaan (sivusuunnan keskelle ja vaakatasoon). Tämä komento on muodostettu liitteen B tietojen perusteella ja on seuraava: 0xff, 0x01, 0x00, 0x07, 0x00, 0x00, 0x08.

6. KEHITETTY OHJELMISTO

Kehitetyn ohjelmiston tarkoituksena on laskea tutkan lähettämän maalitiedon perusteella sivusuunta, korkeussuunta sekä etäisyys kohteeseen. Tämän jälkeen ohjelma kääntää kääntöpään haluttuun suuntaan ja lähettää etäisyystiedon kameran ohjaukselle.



Kuva 21. Eri tietoliikenneyhteyksillä lähetettävät tiedot.

Tutkan palvelin lähettää maalitiedot xml-sanomana TCP-yhteydellä. Esimerkki xml-sanomasta on esitetty liitteessä C.

Kameran ottaman kuvan jatkokäsittelyyn tarvitaan tieto kohteen etäisyydestä. Etäisyystieto lähetetään kameranohjauspalvelimelle UDP-sanomana.

Kääntöpäätä varten lasketaan sivu- ja korkeussuunnat kameran sijainnista kohteen sijaintiin. Sivusuunta ilmoitetaan asteina pohjoisesta niin, että pohjoissuunta on 0° ja asteluku kasvaa suunnan kääntyessä myötäpäivään. Korkeussuunta ilmoitetaan asteina vaakatasosta niin, että vaakataso on 0° ja asteluku kasvaa suunnan kääntyessä ylös ja pienenee suunnan kääntyessä alas. Suuntatietojen perusteella ohjataan kääntöpää oikeaan suuntaan. Ohjauskomennot lähetetään Pelco "D" -protokollalla RS-485 yhteyden kautta.

Kääntöpään testaamiseksi, etäisyyden ja suuntatiedon laskemiseksi sekä kääntöpään ja kameran ohjauspalvelimen ohjaamiseksi tehtiin seuraavat ohjelmat:

- ohjelma, jolla kääntöpäätä voidaan ohjata manuaalisesti eri suuntiin
- ohjelma, joka simuloi tutkan palvelinta lähettämällä xml-sanomia

- ohjelma, joka vastaanottaa edellä mainitun xml-sanoman, parsii siitä tarvittavat tiedot, laskee ennakkopisteen suunnan ja etäisyyden, ohjaa kääntöpään kyseiseen suuntaan ja lähettää etäisyystiedon kameran ohjauspalvelimelle.

6.1 Kääntöpään manuaalinen ohjaus

Kääntöpäätä ohjataan antamalla sille kääntymisnopeus ja kääntymissuunta, johon päätä ajetaan. Testausta varten tehtiin ohjelma, jolla kääntöpäätä voidaan ohjata eri suuntiin. Ohjelman pseudokoodi on esitetty alla.

```

main(argc, argv[]) { //arc=parametrien määrä, argv[]=parametrien arvot
2      //Asetetaan arvot vakiotavuille
      Pleft = '\x04'
4      Pright = '\x02'
      Tup = '\x08'
6      Tdown = '\x10'
      gotohome[] = "\xff\x01\x00\x07\x00\x00\x08"
8      //Asetetaan arvot komennon rungon tavujonolle
      cmd[] = "\xff\x01\x00\x00\x40\x3F\x00"
10
      //Asetetaan sarjaportin parametrit
12     yhteys = aseta_portti(2400 bps, 8 bit, no parity, 1 stop bit)

14     //1 parametri joka on H → ajetaan kotiasemaan
     if ((argc = 2) AND (argv[1] = 'H'))
16         lähetä(yhteys, gotohome, 7)

18     //3 parametria: suunta, aika, nopeus
     if (argc = 4) {
20         spd = argv[3]
         //Nopeusarvo oikeaan paikkaan
22         if (argv[1] = 'U') OR strchr(argv[1], 'D')) cmd[5] = spd
         if (argv[1] = 'L') OR strchr(argv[1], 'R')) cmd[4] = spd
24     }

26     //Jos parametreja 3 tai enemmän
     if (argc >= 3) {
28         //Luodaan komento
         if (argv[1] = 'U') mkcmd(cmd,Tup,1)
30         if (argv[1] = 'D') mkcmd(cmd,Tdown,1)
         if (argv[1] = 'R') mkcmd(cmd,Pright,1)
32         if (argv[1] = 'L') mkcmd(cmd,Pleft,1)
         t = argv[2] //kääntymisaika
34         //Suuntaus
         t_C_ready = aika_nyt + t; //Suuntaus valmis
36         lähetä(yhteys, cmd, 7); //Lähetetään komento
         //Odotetaan, että suuntaus on valmis
38         while (t_C_ready > aika_nyt) {}
         //Luodaan ja lähetetään pysäytyskomento
40         mkcmd(cmd, Tup|Tdown|Pleft|Pright, 0)
         lähetä(yhteys, cmd, 7)
42     }
}

```

Ohjelma 1. Pseudokoodi kääntöpään manuaaliohjaukselle.

Jos samaan aikaan haluttaisiin ajaa kääntöpäätä myös ylös (Tup), kutsuttaisiin `mkcmd`-funktiota komennolla `mkcmd(cmd, Tup OR Pright, 1)`, jolloin `dir`-muuttuja saisi arvon `Tup OR Pright`. Katso kuvan 21 oikea puolisko.

Edellä kuvattu menettely jättää tavun 4 muut bitit koskemattomiksi (bitit joiden arvo muuttujassa `dir` on 0).

Vastaavasti kun `act`-muuttujan arvo on 0, asetetaan ohjelman 1 rivillä 43 muuttujan `dir` arvoksi 0001 1110. Nyt ohjelmassa 2 tavun 4 arvo asetetaan loogisten NOT- ja AND-operaatioiden avulla. Näin saadaan halutut bitit nollattua, mutta muut bitit pysyvät koskemattomina. Katso kuva 22.

Pright	0	0	0	0	0	0	1	0	}	OR
Pleft	0	0	0	0	0	1	0	0		
Tup	0	0	0	0	1	0	0	0		
Tdown	0	0	0	1	0	0	0	0		
<hr/>										
dir	0	0	0	1	1	1	1	0	}	AND
NOT (dir)	1	1	1	0	0	0	0	1		
Tavu 4 vanha-arvo	x	x	x	x	x	x	x	x		
Tavun 4 uusi arvo	x	x	x	0	0	0	0	x		

x on joko 0 tai 1

Kuva 23. Bittitason loogiset NOT- ja AND-operaatiot.

6.2 Tutkan simulointi

Tutkaa simuloitiin yksinkertaisella TCP-palvelimella, joka lähetti liitteessä C kuvattua, xml-sanomaa. Sanoma sisältää tiedot vain yhdestä tutkan antamasta kohteesta.

TCP-palvelimen pseudokoodi on esitetty alla.

```

main() {
2   openTCPsocket(MYPORT)

4   while(uusi_yhteystunnus) { } //Odotetaan yhteyttä asiakkaalta

6   str = luetiedosto("teststream.xml") //Luetaan tiedosto

8   edaika = 0
   while (1) { //Ikuinen silmukka
10    sanoma = etsi ensimmäinen "<Robin>-</Robin>" kehys str-muuttujasta
        poista sanoma str-muuttujasta
12
        if(edaika > 0) {
14            if (sanoma sisältää "<Tracks Timestamp=") {
                    aika = Tracks_Timestamp
16            valmis = aika_nyt + (aika - edaika)
                    while (valmis > aika_nyt) { } //Odotetaan

```

```

18         edaika = aika
19     }
20 }
21 else {
22     if (sanoma_include "<Tracks Timestamp=") {
23         edaika = Tracks_Timestamp
24     }
25     send(uusi_yhteystunnus, sanoma) //Lähetetään sanoma
26     if (str == "") break //Jos str on tyhjä, lopetetaan
27 }
28 }

```

Ohjelma 3. Tutkan xml-sanomasimulaattorin pseudokoodi.

Ohjelman 3 rivillä 2 avataan TCP-portti. MYPORIT vakio määrittää porttinumeron jota kuunnellaan. Rivi 4 on silmukka, jossa odotetaan asiakkaan yhteydenmuodostusta. Rivi 6 lukee tiedostosta halutut sanomat. Rivit 9 – 29 muodostavat ikuisen silmukan, jossa luetaan <Robin> - </Robin> sanomakehykset järjestyksessä ja jokaisen kehyksen Tracks-tagista luetaan kellonaika. Robin-kehykset lähetetään sillä aikavälillä kuin Tracks-tagin kellonajat määrittävät. Kun tiedostosta on lähetetty kaikki Robin-kehykset, lopetetaan ohjelman suoritus.

6.3 Ohjelmarunko laskentafunktioiden kehitykseen

Eri funktioiden kehitystä varten tarvittiin ohjelmarunko, joka vastaanottaa tutkan xml-sanoman, parsii siitä tarvittavat tiedot, laskee suunnan kamerasta kohteeseen ennakkopisteseen, ohjaa kääntöpään ennakkopisteen suuntaan ja lähettää etäisyystiedon eteenpäin.

Ohjelmarunko sisältää siis seuraavat kokonaisuudet:

- TCP-soketin, joka ottaa yhteyden tutkan palvelimeen ja vastaanottaa xml-sanomia
- UDP-soketin, joka lähettää etäisyystiedon
- xml-sanoman parsinnan
- ennakkopisteen suunta- ja etäisyystietojen laskennan
- kääntöpään ohjauksen
- apufunktiot.

Seuraavassa esitetään ohjelmarungon pseudokoodi ja sen jälkeen käsitellään tarkemmin ennakkopisteen etäisyyden ja suunnan laskennan funktiot.

Apufunktioiden osalta esitetään vain laskennan kannalta oleelliset funktiot. Muut apufunktiot kuten esimerkiksi muunnosfunktio asteista radiaaneiksi jätetään esittelemättä.

```

2 main() {
3     Pleft = '\x04', Pright = '\x02', Tup = '\x08', Tdown = '\x10'
4     gotohome[] = "\xff\x01\x00\x07\x00\x00\x08"
5     cmd[] =      "\xff\x01\x00\x00\x40\x3F\x00"
6
7     //Alustus
8     //Kääntöpää mekaanisesti keskisektoriin ja vaakatasoon

```

```

8      az = kääntöpään_maksimi_siirtymä_keskisektoriin
      el = kääntöpään_maksimi_siirtymä_keskisektoriin
10
      //Aika
12      t = az / kameran_sivusuunnan_nopeus
      if (t < (el / kameran_sivusuunnan_nopeus))
14          t = el / kameran_sivusuunnan_nopeus
      //Alustetaan yhteys kääntöpäähän
16      yhteys_COM = aseta_portti(2400 bps, 8 bit, no parity, 1 stop bit)

18      //Aika ja komento kääntöpäälle
      t_C_ready = aika_nyt + t
20      lähetä(yhteys_COM, gotohome)

22      //Odotetaan, että kääntöpää on valmis
      while (t_C_ready > aika_nyt) { } //Odota t_C_ready saakka
24

      //Alustetaan yhteys tutkaan
26      yhteys_TCP = aseta_portti(ip_R, port_R)
      //Alustetaan yhteys kameran ohjaukseen
28      yhteys_UDP = aseta_portti(ip_C, port_C)

30      //Luetaan tutkadata ja lasketaan ennakkopisteen suunta
      for(;;) { //Tutkan lukusilmukka
32          buf = vastaanota(yhteys_TCP) //Tutkadata tutkan palvelimelta

34          if (saatiin uusi xml-sanoma) {

36              //Parsitaan data, josta saadaan
              id_B = kohteen id-numero
38              lat_B = kohteen latitude
              lon_B = kohteen longitude
40              alt_B = kohteen korkeus
              v_B_az = kohteen lentosuunta vaakatasossa
42              v_B_el = kohteen lentosuunta pystytasossa
              v_B = kohteen lentonopeus

44              if (id_B on sama jonka suuntaus on kesken
              TAI suuntaus on valmis)
46                  az_C_old = kääntöpään nykyinen sivusuunta
48                  el_C_old = kääntöpään nykyinen korkeussuunta

50                  //Lasketaan parsitun datan perusteella
                  dist_B = suora etäisyys ennakkopisteeseen
52                  el_B = korotuskulma ennakkopisteeseen
                  az_B = sivukulma ennakkopisteeseen

54                  //Suunta johon pitää kääntää
                  if (el >= el_old) el_dir = Tup
                  else el_dir = Tdown

56                  if (az >= az_old) az_dir = Pright
                  else az_dir = Pleft

58                  //Suuntausaika
                  t_el = abs((el - el_old) /
60                      kääntöpään korkeussuunnan nopeus)
                  t_az = abs((az - az_old) /
62                      kääntöpään sivusuunnan nopeus)

```

```

68         ready_C_az = aika_nyt + t_az
        ready_C_el = aika_nyt + t_el
70         //Komennon muodostus
        mkcmd(cmd, el_dir|az_dir, 1)
72
        //Komento kääntöpäälle
74         lähetä(yhteys_COM, cmd)
        }
76     }
78
    if (aika_nyt >= ready_C_az JA sivusuuntaus kesken) {
80        //Pysäytyskomennon muodostus ja lähetys
        mkcmd(cmd, Pleft|Pright, 0)
82        lähetä(yhteys_COM, cmd)
        az_C_old = az_B //Kameran suunta muistiin
84    }
86
    if (aika_nyt >= ready_C_el JA korotuskulman suuntaus kesken) {
88        //Pysäytyskomennon muodostus ja lähetys
        mkcmd(cmd, Tup|Tdown, 0)
        lähetä(yhteys_COM, cmd)
90        el_C_old = el_B //Kameran suunta muistiin
    }
92
    if (suuntaus valmis JA kuvaa ei ole otettu)
94        lähetä(yhteys_UDP, dist_B)
    }
96 }
}

```

Ohjelma 4. Ohjelmarunko laskentafunktioiden kehitystä varten.

Varsinaisia laskentafunktioita käytetään ohjelmarunon riveillä 50 – 53. Riveillä 55 – 91 toteutetaan kääntöpään ohjaus.

6.4 Etäisyyden laskenta

Kahden pisteen välinen suora etäisyys (E) suorakulmaisessa koordinaatistossa lasketaan matemaattisesta peruskaavasta:

$$E = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (17)$$

jossa Δx , Δy ja Δz ovat kahden pisteen (kohteen sijainti ja kameran sijainti) koordinaattien erotukset.

Koska tutkan lähettämä kohteen sijainti sekä kameran sijainti esitetään WGS84-koordinaattijärjestelmän maantieteellisinä koordinaatteina, muutetaan koordinaatit ensin suorakulmaiseen koordinaatistoon. Tämä tapahtuu luvussa 4 esiteltyjen kaavojen 8, 9, 11, 14 ja 16 avulla. Ennen laskentaa on kohteen ja kameran korkeudet korjattu kohdan 4.4 kaavalla 7. Laskennan pseudokoodi on esitetty ohjelmassa 5.

```

//lat, lon, h --> x,x,x
2 coord(X, Y, Z, fi, la, h) {
    f = 1/ELLIPSOID_INV_F
4    b = ELLIPSOID_A * (1 - f)

6    //Poikittaistaarevuussäde
    N = (ELLIPSOID_A^2) / neliöjuuri(ELLIPSOID_A^2 * cos(fi)^2
8        + b^2 * sin(fi)^2)

10    x = (N+h) * cos(fi) * cos(la)
    y = (N+h) * cos(fi) * sin(la)
12    z = (b^2/ELLIPSOID_A^2 * N + h) * sin(fi)
}

```

Ohjelma 5. *x-, y- ja z-koordinaattien laskenta maantieteellisistä koordinaateista.*

Ohjelmassa 5 ELLIPSOID_F_INV ja ELLIPSOID_A ovat ohjelman otsikkotiedostossa annettuja vakioarvoja ellipsoidin litistysuhteen käänteisarvolle ja pikkuakselin puolikaalle.

Saaduista koordinaateista lasketaan koordinaattien erotukset, minkä jälkeen lasketaan etäisyys kaavasta 17.

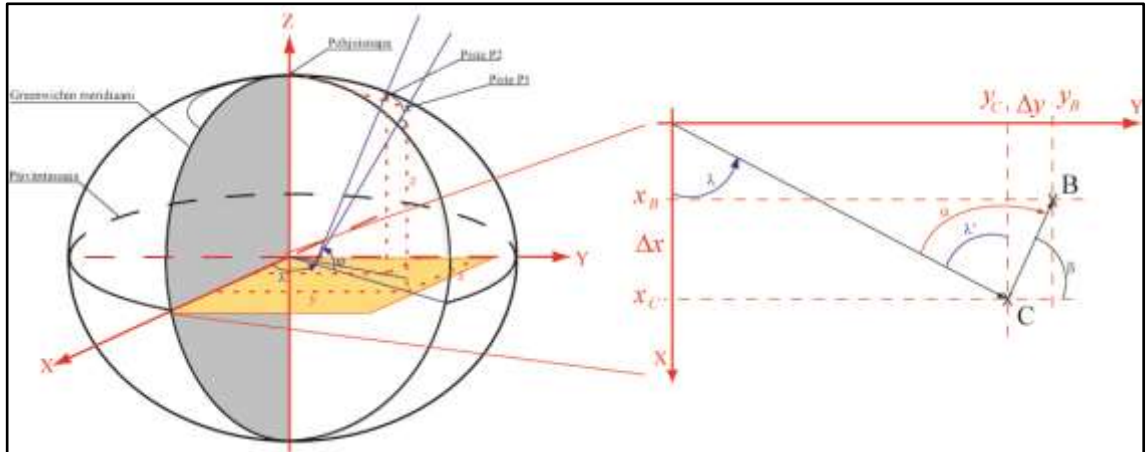
6.5 Sivusuunnan laskenta

Ennakkopisteen sivusuunnan laskemiseksi tehtiin kolme eri funktiota, joiden tarkkuutta oli tarkoitus vertailla lopullisen ohjelmiston testauksen yhteydessä. Sivusuunta ilmoitetaan asteina pohjoisesta, jonka suunta on 0° . Suuntakulma kasvaa käännettäessä myötäpäivään. Pääilmansuuntien asteluvut siis ovat:

- Pohjoinen 0°
- Itä 90°
- Etelä 180°
- Länsi 270° .

Sivusuunnan laskenta XY-projektioitasosta

Kolmiulotteisessa XYZ-koordinaatistossa pohjoissuunta voidaan laskea XY-tason projektioista.



Kuva 24. Kolmiulotteisen XYZ-koordinaatiston XY-projektiotaso.

Vektorin, joka kulkee origosta projektiopisteeseen C (kamera), ja X-akselin välinen kulma, λ , on pisteen C maantieteellisen pituuskoordinaatin arvo (pituuskulma). Tämän saman vektorin vastavektorin suunta kuvaa pohjoissuuntaa pisteestä C (suuntakulma 0°). Laskemalla kulma α , saadaan suuntakulma pisteestä C pisteeseen B (kohde).

Geometriaan perustuen kulma λ ja λ' ovat yhtä suuret (yhdenmuotoiset kolmiot). Trigonometrian perusteella voidaan laskea kulman β arvo:

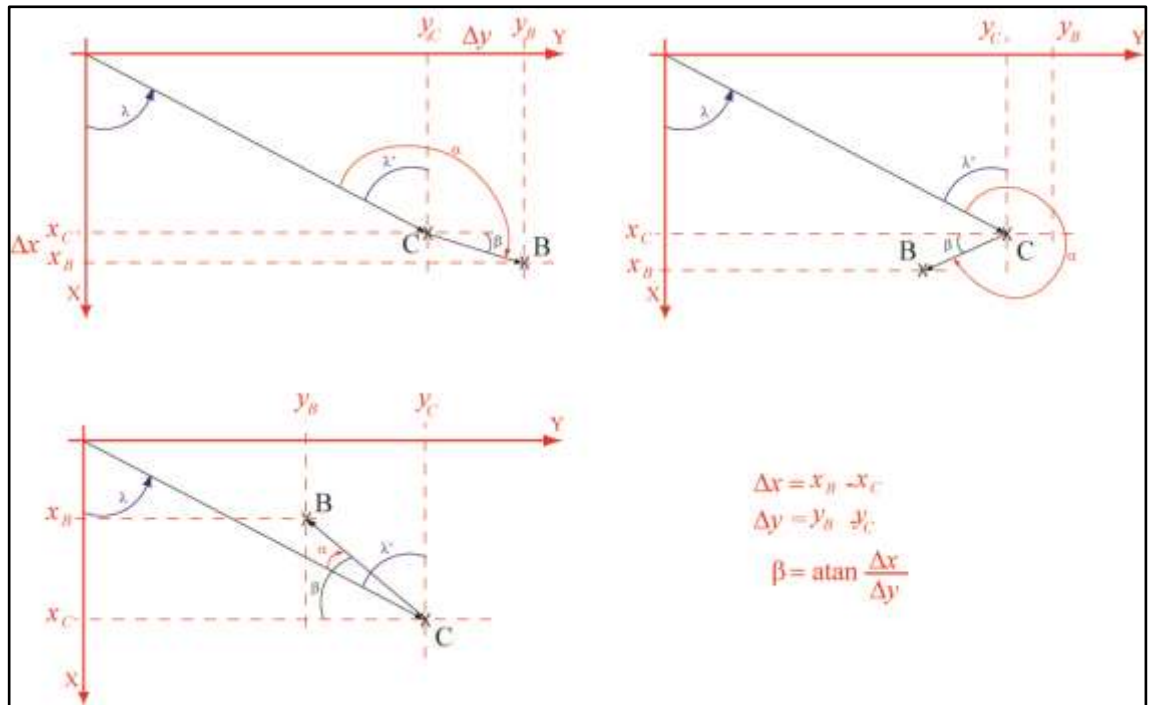
$$\beta = \arctan \frac{\Delta x}{\Delta y}. \quad (18)$$

Koska trigonometrinen arkustangentti-funktio on määritelty siten, että se saa arvoja välillä $-90^\circ \dots 90^\circ$ riippuen argumentin suuruudesta ja etumerkistä, pitää tarkastella tapauskohtaisesti tilanteet, joissa kohde on C-keskeisen koordinaatiston eri neljänneksissä. Kohteen sijainti I neljänneksessä on esitetty kuvassa 24 ja silloin arkustangentti-funktion argumentti on negatiivinen ja sen palauttaman kulman arvo on myös negatiivinen, saadaan kulma α kaavasta:

$$\alpha = \beta - (90^\circ - \lambda) = \lambda + \beta - 90^\circ \quad (19)$$

jossa vakio 90° on neljänneksestä riippuva korjaus.

Neljänneksien II - IV tapaukset on esitetty kuvassa 25.



Kuva 25. Suuntakulma kun kohde sijaitsee eri neljänneksissä.

II-neljänneksessä (kuva 25, vasen yläkulma) kulma β saa positiivisen arvon ja suuntakulma saadaan kaavasta:

$$\alpha = \lambda + 90^\circ + \beta = \lambda + \beta + 90^\circ. \quad (20)$$

III-neljänneksessä (kuva 25, oikea yläkulma) kulma β saa negatiivisen arvon ja suuntakulma saadaan kaavasta:

$$\alpha = \lambda + 270^\circ + \beta = \lambda + \beta + 270^\circ. \quad (21)$$

IV-neljänneksessä (kuva 25, vasen alakulma) kulma β saa jälleen positiivisen arvon ja suuntakulma saadaan kaavasta 21.

λ voi suomessa saada vain positiivisia alle 90° arvoja (Suomi sijaitsee pituuskuulmien 19° ja 32° välissä). β taas, arkustangentin määrittelyn perusteella, saa arvoja välillä $-90^\circ \dots 90^\circ$. Tällöin α voi saada kaavassa 21 arvon, joka on $\geq 360^\circ$. Mikäli näin käy, vähennetään siitä 360° , jolloin saadaan oikea suuntakulma.

Kun tarkastellaan kaavaa 19, havaitaan, että ottamalla huomioon IV-neljänneksen lisärajoite yli 360° kulmasta, voidaan myös se kirjoittaa kaavan 21 muotoon.

Vielä on tarkasteltava tilanne, jossa Δy on nolla, koska jakaminen nolllalla ei ole mahdollista. Jos Δy on nolla ja Δx on positiivinen, on kulma $\beta = -90^\circ$, jolloin suuntakulma voidaan laskea kaavasta 21. Jos Δx on negatiivinen, on kulma $\beta = 90^\circ$ ja suuntakulma voidaan laskea kaavasta 19 tai kaavasta 21 kun huomioidaan 360° rajoitus.

Yhteenveto kaavoista on koottu taulukkoon 6.

Taulukko 6. *XY-tasosta lasketun suuntakulman laskentakaavan korjaus.*

Δx :n etumerkki	Δy :n etumerkki tai arvo	β :n etumerkki tai arvo	$(\lambda + \beta)$:n korjaus
+	+	+	90°
+	-	-	270°
-	+	-	90°
-	-	+	270°
+	0	-90°	90°
-	0	90°	270°

Koordinaattimuunnos maantieteellisistä koordinaateista suorakulmaiseen koordinaatistoon tehdään kohdan 6.4. ohjelmalla 5.

Suunnan laskennan pseudokoodi on esitetty ohjelmassa 6.

```

//Suunta XY-projektiosta
2 bearing(ref_lon, X1, Y1, X2, Y2) {
    beta = 0
4     dX = X2 - X1
    dY = Y2 - Y1
6     if (dY > 0) corr = PI/2 //90 ast
    else if (dY < 0) corr = 3*PI/2 //270 ast
8     else {
        if (dX >= 0) {
10         beta = -PI/2
            corr = PI/2
12         }
        else {
14         beta = PI/2
            corr = 3*PI/2
16         }
    }
18
    if (beta = 0) beta = atan(dX/dY)
20    alfa = ref_lon + beta + corr
22
    if (alfa >= 2*PI) alfa = alfa - 2*PI
24    return alfa
}

```

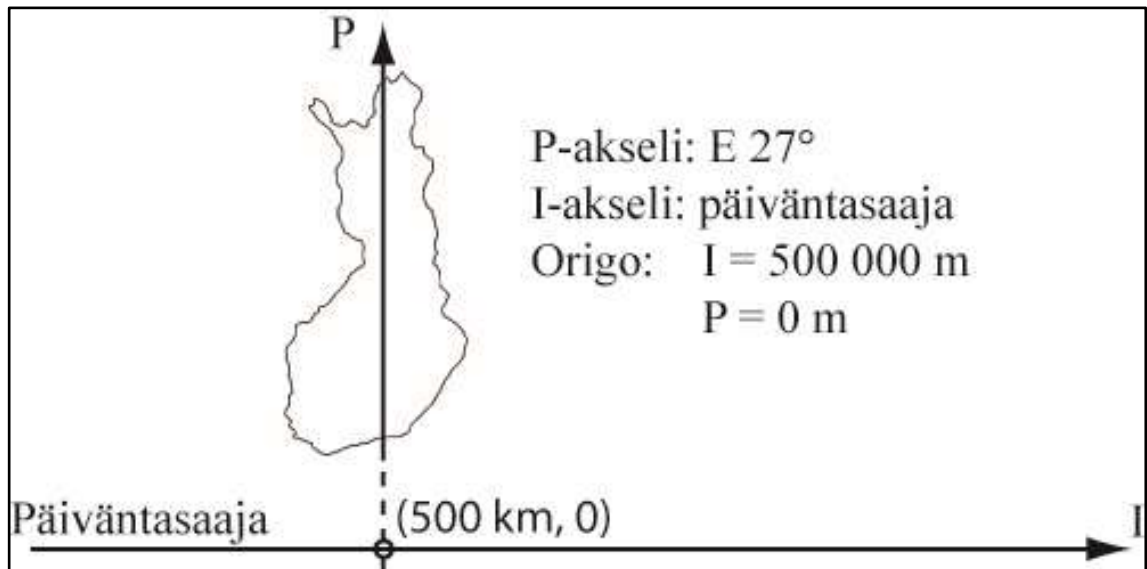
Ohjelma 6. *Suuntakulman laskenta XY-projektiosta.*

Ohjelma 6 laskee annetuista parametreista suuntakulman, jonka se palauttaa. Parametri ref_lon on saatu tutkadatasta ja parametrit X1, Y1, X2 ja Y2 on laskettu maantieteellisistä koordinaateista ohjelmalla 5. PI on piin likiarvo, joka on määritetty otsikkotiedostossa.

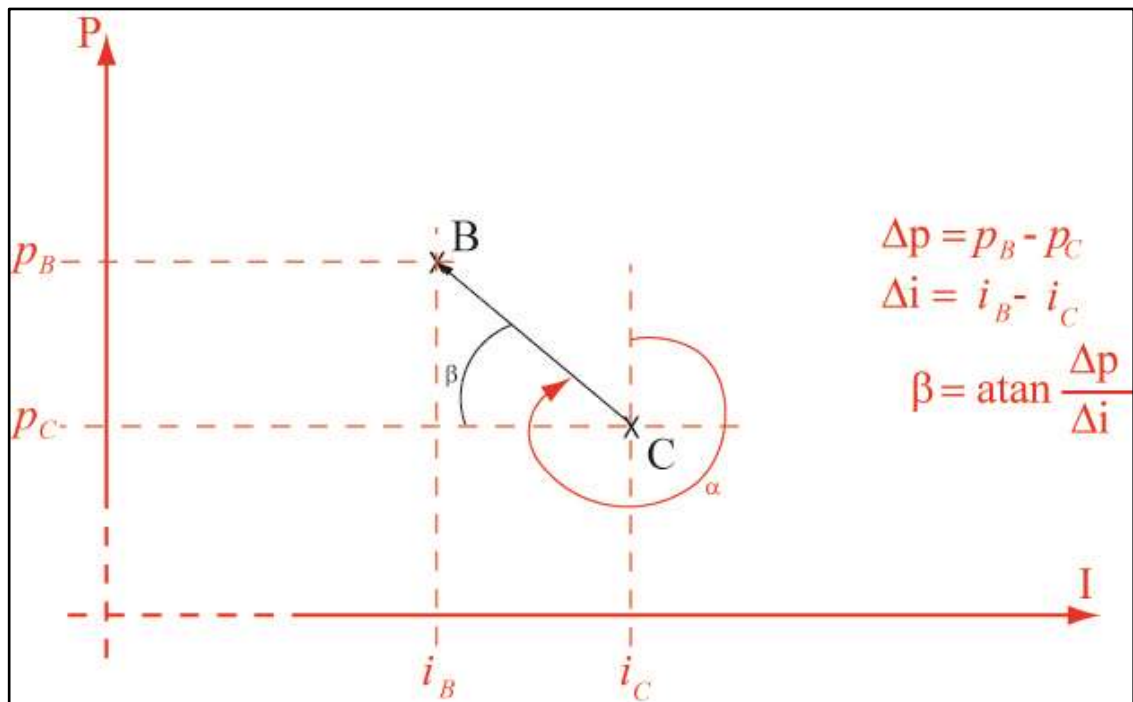
Sivukulman laskenta PI-projektiotasosta

Laskennassa kaksi pistettä projisoidaan ensin karttatasolle, katso kohta 4.1, kuva 8, jolloin saadaan niiden P- ja I-koordinaatit. Kuten kohdassa 4.3 esitetään, Suomi asettuu koordinaatistoon kuvan 26 mukaisesti. Koska karttakoordinaatisto on suorakulmainen, voidaan suuntakulma laskea kulmana P-akseliin nähden.

Laskentaperiaate on esitetty kuvassa 27.



Kuva 26. PI-tasokoordinaatisto.



Kuva 27. Suuntakulman laskeminen PI-tasokoordinaatistossa.

Kun tehdään sama tarkastelu kuin XY-projektion tapauksessa saadaan kulman α kaavaksi I- ja II-neljänneksessä:

$$\alpha = 90^\circ - \beta \quad (22)$$

ja III-, sekä IV-neljänneksessä:

$$\alpha = 270^\circ - \beta. \quad (23)$$

Kaavoissa 22 ja 23 kulma α pysyy kaikilla mahdollisilla β :n arvoilla välillä $0^\circ \dots 360^\circ$.

Kun Δi on nolla, saa β arvon 90° , kun Δp on positiivinen ja arvon -90° , kun Δp on negatiivinen.

Yhteenvedo kaavoista on koottu taulukkoon 7.

Taulukko 7. *PI-tasosta lasketun suuntakulman laskentakaavan korjaus.*

ΔP :n etumerkki	ΔI :n etumerkki tai arvo	β :n etumerkki tai arvo	β :n korjaus
+	+	+	90°
+	-	-	270°
-	+	-	90°
-	-	+	270°
+	0	90°	90°
-	0	-90°	270°

Suunnan laskennan pseudokoodi on esitetty ohjelmassa 7.

```

2 //Lasketaan suunta pisteestä 1 pisteeseen 2
bearing(fi1, fi2, la1, la2) {
    //Tasokoordinaatit
4    coordNE(P1, I1, fi1, la1);
    coordNE(P2, I2, fi2, la2);
6
    //Koordinaattierot
8    dP = P2 - P1
    dI = I2 - I1
10
    //Suunta 1 --> 2
12    if (dI > 0) corr = PI/2 //90 ast
    else if (dI < 0) corr = 3*PI/2 //270 ast
14    else {
        if (dP >= 0) {
16        beta = PI/2
        corr = PI/2
18        }
        else {
20        beta = -PI/2
        corr = 3*PI/2
22    }
}

```

```

24     }
    if (beta = 0) beta = atan(dP/dI)
26     alfa = corr - beta
28     return alfa
}

```

Ohjelma 7. *Suuntakulman laskenta PI-projektioista.*

Ohjelman 7 parametrit fi_1 ja la_1 ovat kameran maantieteelliset koordinaatit. fi_2 ja la_2 ovat kohteen maantieteelliset koordinaatit. P_1 -, P_2 -, I_1 - ja I_2 -koordinaatit lasketaan funktiossa *coordNE()*. Funktio käyttää JUHTA:n (julkisen hallinnon tietohallinnon neuvottelukunta) suosituksessa JHS (julkisen hallinnon suositus) 154 määrittämiä muunnoskaavoja, jotka on esitetty liitteessä F.

P- ja I-koordinaattien laskennan pseudokoodi on esitetty ohjelmassa 8.

```

//Koordinaattien laskenta PI-projektiotasolle
2 coordNE(P, I, fi, la) {
    f = 1/ELLIPSOID_INV_F //ellipsoidin litistysuhde
4    n = f/(2-f) //ellipsoidin toinen litistysuhde
    e = neliöjuuri(2 * f - f^2) //ellipsoidin eksentrisyys
6    h1p = n/2 - 2 * n^2 / 3 + 5 * n^3 / 16 + 41 * n^4 / 180
    h2p = 13 * n^2 / 48 - 3 * n^3 / 5 + 557 * n^4 / 1440
8    h3p = 61 * n^3 / 240 - 103 * n^4 / 140
    h4p = 49561 * n^4 / 161280
10
    //meridiaanin pituisen ympyrän säde
12    A1 = ELLIPSOID_A*(1 + n^2 / 4 + n^4 / 64) / (1 + n)
    Qp = asinh(tan(fi))
14    Qpp = atanh(e * sin(fi))
    Q = Qp - e*Qpp
16    l = la - MAP_LAMBDA0
    beta = atan(sinh(Q))
18    netap = atanh(cos(beta) * sin(l))

20    epsilonp = asin(sin(beta) * cosh(netap))
    epsilon1 = h1p * sin(2 * epsilonp) * cosh(2 * netap)
22    epsilon2 = h2p * sin(4 * epsilonp) * cosh(4 * netap)
    epsilon3 = h3p * sin(6 * epsilonp) * cosh(6 * netap)
24    epsilon4 = h4p * sin(8 * epsilonp) * cosh(8 * netap)

26    neta1 = h1p * cos(2 * epsilonp) * sinh(2 * netap)
    neta2 = h2p * cos(4 * epsilonp) * sinh(4 * netap)
28    neta3 = h3p * cos(6 * epsilonp) * sinh(6 * netap)
    neta4 = h4p * cos(8 * epsilonp) * sinh(8 * netap)
30
    epsilon = epsilonp + (epsilon1 + epsilon2 + epsilon3 + epsilon4)
32    neta = netap + (neta1 + neta2 + neta3 + neta4)

34    P = A1 * epsilon * MAP_K0
    I = A1 * neta * MAP_K0 + MAP_E0
36 }

```

Ohjelma 8. *PI-koordinaattien laskenta maantieteellisistä koordinaateista.*

Ohjelman 8 saa syötteenä maantieteelliset koordinaatit ϕ_1 ja λ_1 . Ohjelma laskee ja palauttaa koordinaatit P ja I . Vakiot MAP_LAMBDA0, MAP_K0 ja MAP_E0 on määritetty ohjelmiston otsikkotiedostossa ja ovat projektion keskimeridiaani, keskimeridiaanin mitakaavakerroin ja I-koordinaatin arvo keskimeridiaanilla.

Sivukulman laskenta loxodromipurjehduksen purjehdussuunnan avulla

Loxodromipurjehduksessa purjehdussuunta on vakio. Loxodromi on käyrä, joka leikkaa kaikki maapallon meridiaanit vakiokulmassa. Meridiaanit taas ovat maapallon pohjois-eteläsuunnassa kulkevia linjoja, joiden pituuskulman arvo on vakio. (Alexander, 2004, s. 349)

Purjehdussuuntaa voidaan siis tässä tapauksessa käyttää suuntakulmana kamerasta (lähtöpiste) kohteeseen (loppupiste).

Suunnan laskennassa käytetään Kaukorannan esittelemää kaavaa, joka laskee purjehdussuunnan K (Kaukoranta, 2006, s. 1).

$$K = \arctan \left(\frac{\lambda_2 - \lambda_1}{\ln \left(\frac{\tan(\frac{\pi + \phi_2}{4})}{\tan(\frac{\pi + \phi_1}{4})} \right) - a \cdot (\sin \phi_2 - \sin \phi_1) - b \cdot (\sin^3 \phi_2 - \sin^3 \phi_1) - c \cdot (\sin^5 \phi_2 - \sin^5 \phi_1)} \right) \quad (24)$$

missä:

$$a = e^2 \approx 0.00669437999013 \quad (25)$$

$$b = \frac{e^4}{3} \approx 1.4938241150749 \cdot 10^{-5} \quad (26)$$

$$c = \frac{e^6}{5} \approx 6.00013575883841 \cdot 10^{-8} \quad (27)$$

Kaavan 20 kaikki kulmat on esitettävä radiaaneina ja alaindeksi 1 viittaa lähtöpisteeseen sekä alaindeksi 2 loppupisteeseen. Kaavoissa 25-27 e on ellipsoidin ensimmäinen eksentrisyys.

Pseudokoodi suunnan laskennalle on esitetty ohjelmassa 9.

```
//Loxodromi
2 bearing(fi1, fi2, la1, la2) {
    printf("bearing_0\n")
4     f = 1/ELLIPSOID_INV_F //ellipsoidin litistysuhde
    e = neliöjuuri(2 * f - f^2) //ellipsoidin eksentrisyys
6     //Kulmaerot
    dla = la2 - la1
8     dfi = fi2 - fi1

10     L = log(tan(PI/4 + fi2/2) / tan(PI/4 + fi1/2)) //Kaavan log-lauseke
```

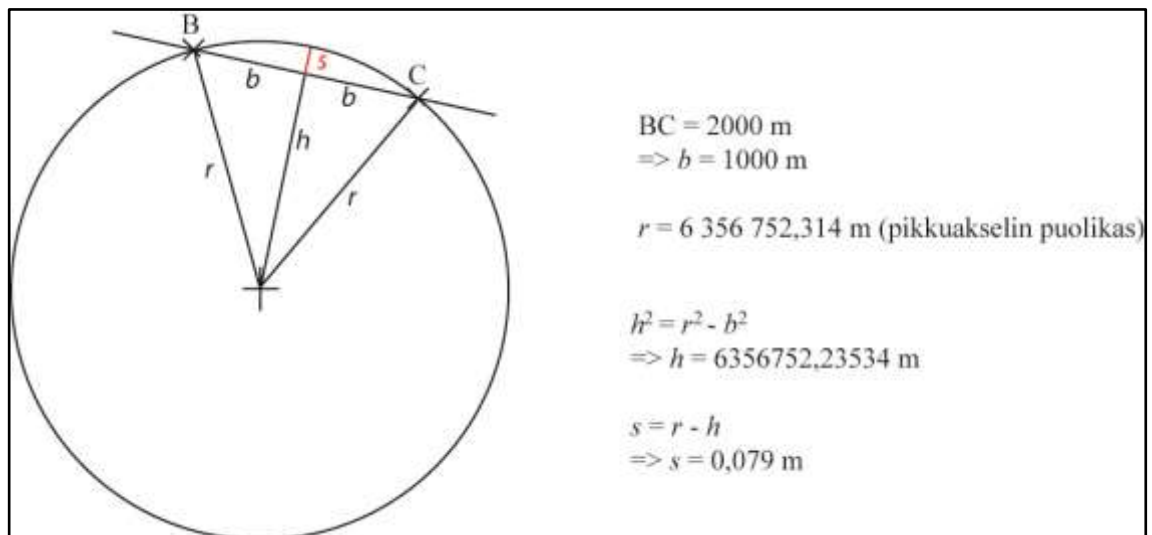
```

12      //sin termien kertoimet (korjaus pallopinnalta => ellipsiin)
      a = e^2
14      b = e^4 / 3
      c = e^6 / 5
16
      //Suunta
18      bear = atan(dla /
20                  (L - a*(sin(fi2) - sin(fi1))
21                  - b*(sin(fi2)^3 - sin(fi1)^3)
22                  - c*(sin(fi2)^5 - sin(fi1)^5)
23
24      //Korjataan kulma erotuksen mukaan
25      if (dfi < 0) bear = bear + PI
26      else if (dla < 0) bear = bear + 2*PI
27
28      return bear
    }

```

Ohjelma 9. Suuntakulman laskenta loxodromipurjehduksen avulla.

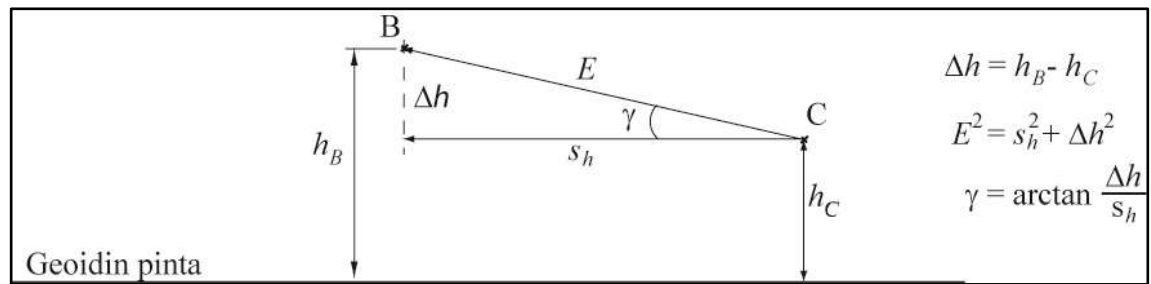
6.6 Korkeuskulman laskenta



Kuva 28. Maapallon kaarevuuden tarkastelua.

Kohteen ja kameran etäisyys on maksimissaan noin 2000 m. Jos oletetaan maapallon säteeksi ellipsoidin pikkuakselin puolikas, jolloin pinnan kaarevuus on maksimissaan, ja lasketaan näin muodostuneen ympyrän janteen korkeus, s , saadaan sen arvoksi noin 8 cm. Katso kuva 28. Kaarevuus on niin pieni, että olettamalla maapallon (geoidin) pinta suoraksi ei tehdä merkittävää virhettä korkeuden laskemisessa.

Kohteen korkeus ja korkeuskulma lasketaan kuvan 29 osoittamalla tavalla.



Kuva 29. Kohteen korkeuden ja korkeuskulman laskentaperiaate.

Koska korkeuseron etumerkki ilmaisee onko kohde kameraa korkeammalla (+) vai matalammalla (-), tulee myös korkeuskulman etumerkki vastaavasti: vaakatason yläpuolella positiivinen (+) ja vaakatason alapuolella negatiivinen (-).

Suora etäisyys E lasketaan kohdan 6.4 kaavasta 17. Horisontaalinen etäisyys s_h lasketaan ratkaisemalla se kuvan 29 mukaisesti.

Korkeuskulma lasketaan ohjelmassa osana funktiota, joka laskee ennakkopisteen sivu- ja korkeuskulmat sekä etäisyyden. Funktion pseudokoodi on esitetty ohjelmassa 10.

6.7 Ennakkopisteen laskenta

Ennakkopisteen laskenta perustuu iterointiin. Ennakkopiste lasketaan seuraavasti:

1. Ennakkopiste on kohteen sijainti.
2. Lasketaan tutkan antamasta kohteen nopeudesta ja suunnasta kohteen pysty- ja vaakasuunnan nopeuskomponentit (v_speed ja h_speed).
3. Lasketaan vaakasuunnan nopeuskomponentista pohjois- ja itäsuuntaiset komponentit (n_speed ja e_speed).
4. Lasketaan kauanko kääntöpään kestää kääntyä nykyisestä asemasta ennakkopisteeseen ($steeringtime_az$ ja $steeringtime_el$).
5. Lasketaan pidemmän kääntymisajan ja kohteen nopeuskomponenttien perusteella, mihin kohde on siirtynyt kääntymisen aikana, tämä on uusi ennakkopiste.
6. Toistetaan kohtia 4 ja 5 kunnes ero uuden- ja edellisen ennakkopisteen välillä on riittävän pieni.

Taulukossa 8 on esimerkki sivukulman iteroinnin lukukarvojen kehittymisestä. Ennakkopiste 4 on lopullinen sivukulma, johon kääntöpää käännetään. Sarakeotsikot viittaavat ohjelman 10 muuttujien nimiin.

Taulukko 8. Ennakkopisteen iteroinnin kehittyminen.

	Käsiteltävä piste				steeringtime	n:o	Ennakkopiste	
	n_distance	e_distance	target_az	d			e_distance	i_distance
Kohde	-814,1 m	-44,02 m	183,095°	-87,905	9,3 s	1	-721,1 m	-205,5 m
Ep 1	-721,1 m	-205,52 m	195,909°	-74,091	8,0 s	2	-733,8 m	-183,3 m
Ep 2	-733,8 m	-183,33 m	194,026°	-75,974	8,2 s	3	-731,96 m	-186,59 m
Ep 3	-731,96 m	-186,59 m	194,301°	-75,699	8,2 s	4	-732,23 m	-186,11 m
Ep 4	-732,23 m	-186,11 m	194,261°	-75,739				

Ennakkopisteen laskenta tehdään osana funktiota, joka laskee ennakkopisteen sivu- ja korkeuskulmat sekä etäisyyden. Funktion pseudokoodi on esitetty ohjelmassa 10.

```

//Lasketaan kameran sivu- ja korkeussuunta
2 panTilt(target_el, target_az, steeringtime_az, steeringtime_el,
    distance, target_lat, target_lon, target_x, target_y, target_z,
4    target_h, ref_lat, ref_lon, ref_x, ref_y, ref_z, ref_h,
    el_old, az_old, ele, azi, spd) {
6
    //Etäisyys kohteeseen
8    dH = target_h - ref_h //Kohteen ja kameran korkeusero
    //Suora etäisyys
10    distance = neliöjuuri((ref_x-target_x)^2 + (ref_y-target_y)^2
        + (ref_z-target_z)^2)
12    h_distance = neliöjuuri(distance^2 - dH^2); //Horisontaali etäisyys
14
    //Suunta kohteeseen
    target_az = bearing() //Sivukulman laskenta
16    target_el = atan(dH/h_distance)
18
    //Etäisyys kohteeseen pohjois- ja itä-suunnassa
    n_distance = cos(target_az) * h_distance
20    e_distance = sin(target_az) * h_distance
22
    //Kohteen horisontaali nopeus
    h_speed = cos(ele) * spd
24    // Kohteen horisontaali nopeus pohjois- ja itä-suuntiin
    n_speed = cos(azi) * h_speed
26    e_speed = sin(azi) * h_speed
28
    //Vertikaalinopeus
    v_speed = sin(ele) * spd
30
    //Etäisyydet ja korkeusero muistiin iterointia varten
32    n_distance_0 = n_distance
    e_distance_0 = e_distance
34    dH_0 = dH
36
    //Apumuuttujat lentomatkoille
    n_flen = 0, e_flen = 0, v_flen = 0
38    steeringtime_az = 0; steeringtime_el = 0
    for(;;) { //Iterointi
40        //Suunnat muistiin
        target_az_old = target_az, target_el_old = target_el

```

```

42
43     //Kääntöpään suntausaika
44     //Az
45     d = target_az - az_old //Suuntaero: - = L, + = R
46     steeringtime_az = d/dtor(HEAD_AZ_SPEED)
47
48     //El
49     d = target_el - el_old //Suuntaero: - = D, + = U
50     steeringtime_el = d/dtor(HEAD_EL_SPEED)
51
52     //Suuntauksen max aika
53     steeringtime = itseisarvo(steeringtime_az)
54     if (itseisarvo(steeringtime_el) > steeringtime)
55         steeringtime = itseisarvo(steeringtime_el)
56
57     //Horisontaali matkat kääntymisen aikana
58     n_flen = steeringtime * n_speed //Pohjoiseen
59     e_flen = steeringtime * e_speed //Etelään
60
61     //Vertikaali matka kääntymisen aikana
62     v_flen = steeringtime * v_speed
63
64     //Uusi paikka
65     //Etäisyys
66     n_distance = n_distance_0 + n_flen
67     e_distance = e_distance_0 + e_flen
68     dH = dH_0 + v_flen
69     h_distance = neliöjuuri(e_distance^2 + n_distance^2)
70     distance = neliöjuuri(h_distance^2 + dH^2) //Suora etäisyys
71
72     //Az
73     target_az = atan(e_distance/n_distance)
74     if (n_distance < 0) target_az += PI
75     else if (e_distance < 0) target_az += 2*PI
76
77     //El
78     target_el = atan(dH/h_distance)
79
80     //Jos tarkkuus riittävä, keskeytetään iterointisilmukka
81     if ((itseisarvo(target_az_old - target_az)
82         < ITERATION_ACCURACY_AZ) AND
83         (itseisarvo(target_el_old - target_el)
84         < ITERATION_ACCURACY_EL)) break
85     }
86 }

```

Ohjelma 10. Ennakkopisteen sivu- ja korkeuskulman laskenta.

6.8 Lähtötietojen tarkkuus

Suunnitelluissa testeissä horisontaalitutkan antaman paikkatiedon käytön ehdoksi suunniteltiin, että leveys- ja pituuskulmat tulee vastaanottaa tutkalta neljän desimaalin tarkkuudella. Mikäli tarkkuus on pienempi, kohde hylätään.

Kaukoranta esittelee loxodromietäisyyden D laskemiseksi merimaileina kaavan, jossa ϕ annetaan minuutteina (Kaukoranta, 2006, s. 2).

$$D = \frac{\phi_2 - \phi_1}{\cos K} \quad (28)$$

Tarkastellaan taulukon 8 kohdetta, joka on suunnassa $183,095^\circ$. Tämä arvo valittiin, koska suunta on laskettu tutkan ja kameran käyttöpaikalla diplomityötä tehtäessä. Jos nyt kuvitellaan, että kamera liikkuu $0,0001^\circ$ tähän suuntaan, voidaan matka laskea muuntamalla ensin asteet minuuteiksi ja laskemalla sitten $0,0001^\circ$ etäisyys merimaileina:

$$0,0001^\circ \cdot 60' / 1^\circ = 0,006'$$

$$D = \frac{0,006}{\cos 183,095} = -0,006$$

Merimailin pituus on keskimäärin 1852,22 m, joten D metreinä on:

$$0,006 \cdot 1852 \text{ m} \approx 11 \text{ m}.$$

Jotta kohteet pystyttäisiin erottamaan toisistaan tällä tarkkuudella pitää tutkan pulssinpitäuden olla kaavan 4 mukaisesti:

$$\delta_R = \frac{\tau \cdot c}{2} \Leftrightarrow \tau = \frac{2 \delta_R}{c} = \frac{2 \cdot 11 \text{ m}}{3 \cdot 10^8 \frac{\text{m}}{\text{s}}} = 73 \cdot 10^{-9} \text{ s} = 73 \text{ ns}$$

Edellä mainitussa testissä kohde suunniteltiin otettavaksi mukaan laskentaan, jos se on maksimissaan 1500 m etäisyydellä. Edellä lasketun perusteella $0,0001^\circ$ tarkkuus edellyttää 11 m kulmaerottelukyvyyä.

Kaavasta 5 saadaan keilanleveyden lausekkeeksi:

$$\theta_{3dB} [^\circ] = \frac{180^\circ}{\pi R} \cdot \delta_A = \frac{180^\circ}{\pi \cdot 1500 \text{ m}} \cdot 11 \text{ m} = 0,42^\circ$$

Tämä on siis keilanleveys joka vaaditaan, jos 11 m kulmaerottelukyvyn vaatimus asetetaan 1500 m etäisyydelle tutkasta. Käytännössä keilanleveyden tulisi olla huomattavasti pienempi, koska kaavan 5 käytännön erottelukyvyn kerroin on 1-1,5.

7. YHTEENVETO JA JATKOKEHITYS

Diplomityön päämääränä oli selvittää, kyetäänkö tietyn 3D-tutkan paikkatiedon avulla suuntaamaan järjestelmäkamera riittävällä tarkkuudella kohteeseen. Kysymys jakautui neljään osakokonaisuuteen, jotka omalta osaltaan vaikuttavat lopputulokseen: suuntauksen tarkkuus, tutkan antaman paikkatiedon tarkkuus, kohteen liiketilan muuttuminen ja laskennan tarkkuus.

Kohdan 3.2 mittauksissa kääntöpään kääntymisnopeus todettiin riittäväksi ja kameran massalla ei havaittu olevan vaikutusta kääntymisnopeuteen ja asettumistarkkuuteen. Suunnan laskenta riittävällä tarkkuudella on teknisesti mahdollista. Rajoituksia siihen aiheuttaa käytännössä vain lähtötietojen tarkkuus ja ennakkopisteen osalta kohteen liiketilan säilyminen.

Mittausten ja testien perusteella ei kuitenkaan voida lopullisesti sanoa, onko kääntöpään suuntaaminen mahdollista riittävällä tarkkuudella. Käytetyn kääntöpään asettumistarkkuus toistuvissa liikkeissä ei mahdollista jatkuvaa toimintaa. Tulee joko sallia kääntöpään suunnan toistuva kalibrointi, jolloin kuvauskohteita todennäköisesti menetetään, tai korvata kääntöpää tekniikalla, jonka suuntauksen toistettavuus on huomattavasti parempi.

Kohdassa 6.8 tutkittiin lähtötietojen tarkkuuksien vaatimuksia. Lisätutkimusaiheena tulisi tarkastella, onko vaadittu etäisyys- ja kulmamittaustarkkuus saavutettavissa käytössä olevilla horisontaali- ja vertikaalitutkillla.

Lopullista sovellusta kehitettäessä tulee myös selvittää, miten kääntöpään toimintaan vaikuttavat esimerkiksi alhaiset lämpötilat.

Myöskään pelkkien maksiminopeuksien käyttö suuntauksessa ei liene järkevää. Noin 6 kg kameran massan aiheuttama hitaus ja nopeiden liiketilan muutoksien aiheuttamat kiihtyvyydet aiheuttavat vaatimuksia kääntöpään ja kameran kiinnityksille ja saattavat myös aiheuttaa ongelmia kameran toimintaan.

Kohteen tunnistamiseksi saattaa olla tarpeen myös seurata kohdetta kääntöpäällä, jolloin kääntöpään kääntymisnopeus tulisi saada sovitettua kohteen liikkeeseen. Tällöin nykyisellä kääntöpäällä ohjelman tulee laskea kääntymisnopeuden ja ennakkopisteen sijainnin perusteella kääntymisaika nopealle käännölle kohteeseen, ja tämän jälkeen uusi nopeus seuranta varten. Koska kääntöpäältä ei nyt saada mitään takaisinkytkentää, perustuu suuntaus pelkästään oletetun nopeuden ja ajan perusteella laskettuun kulmatietoon. Mikäli kääntöpäältä saataisi takaisinkytkentänä todellinen sivu- ja korkeuskulma, helpottuisi suuntauksen hallinta merkittävästi. Takaisinkytkentä poistaisi myös kääntöpään suunnan

kalibroinnin tarpeen jatkuvassa toiminnassa. Edullisia kääntöpäitä tällaisella ominaisuudella ei löydetty, mutta takaisinkytkennän rakentaminen olemassa olevaan kääntöpäähän on myös varteen otettava vaihtoehto.

Kunhan edellä esitettyihin haasteisiin löydetään ratkaisut, on järjestelmäkameran suuntaaminen kohteeseen mahdollista. Kehittämällä nykyisiä laitteita ja ohjelmistoja sekä etsimällä käyttötarkoitukseen paremmin sopivaa tekniikkaa, on haasteet mahdollista ratkaista jo lähitulevaisuudessa. Erilaisia teknisiä ratkaisuja on tarjolla ja tekniikka myös kehittyy huimaa vauhtia.

LÄHTEET

Alexander, J. (2004) ”Loxodromes : A Rhumb Way to Go”, *Mathematics Magazine*, 77(5), ss. 349–356. Saatavissa: <https://search.proquest.com/docview/229825891/fulltextPDF/26DC3EB1BFDA40C6PQ/1?accountid=27303>.

Hooijberg, M. (2008) *Geometrical Geodesy*. doi: 10.1007/978-3-540-68225-7.

Häkli, P., Puupponen, J. ja Koivula, H. (2009) *Suomen geodeettiset koordinaatistot ja niiden väliset muunnokset*.

Kaukoranta, J. (2006) *Loxodrome (Mercator) course and distance on the WGS ellipsoid*. Saatavissa: <http://jukaukor.mbnet.fi/navimat.html>.

Komatinen, P. (2011) *Monitoring of Bird Movement via Marine Radar in the Western Basin of Lake Erie*.

Kozierok, C. M. (2005) ”The TCP/IP Guide : a comprehensive, illustrated internet protocols reference”, *Book*, s. 1648 (est.).

Laurila, P. (2012) *Mittaus- ja kartoitustekniikan perusteet*. 4. uudiste. Rovaniemen ammattikorkeakoulu.

Mackay, S., Wright, E., Reynders, D. ja Park, J. (2004) *Practical Industrial Data Networks*.

Motiva OY (2015) *Tuulivoimaopas - Linnut, Tuulivoimalat ja linnut*. Saatavissa: http://www.tuulivoimaopas.fi/tuulivoiman_ymparisto-_ja_muut_vaikutukset/linnut (Viitattu: 31. heinäkuuta 2017).

Olli, K. ja Lehto, A. (1999) *Tutkatekniikka*. 2. painos. Otatieto.

Pihlaja, M. (2012) *ETELÄ-SAVON TUULIVOIMAVAIHEMAAKUNTAKEAVAN LINNUSTO- JA LEPAKKOSELVITYS*. Saatavissa: www.esavo.fi/resources/public/media/ES_Lintu_ja_lepakko_raportti_lopullinen.pdf.

Poutanen, M. (2000) ”Koordinaatit, korkeus, kartat ja GPS”, ss. 1–17.

Richards, M. A., Scheer, J. A., Holm, W. A., Drive, P. ja Suite, B. (2008) *Principles of Modern Radar Volume I - Basic Principles*. SciTech Publishing. doi: 10.1049/SBRA503E.

Stevens, W. R. (1993) *TCP / IP Illustrated , Volume 1, The Protocols*. Pearson.

Stevens, W. R., Fenner, B. ja Rudoff, A. M. (2003) *UNIX Network Programming, Communication*. Addison Wesley. Saatavissa: <https://books.google.fi/books?id=ptSC4LpwGA0C&printsec=frontcover&hl=fi#v=onepage&q&f=false>.

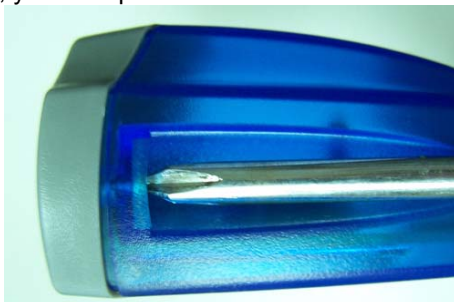
Suomen Tuulivoimayhdistys (2017) *Tuulivoima Suomessa - Suomen Tuulivoimayhdistys, Tuulivoima Suomessa*. Saatavissa: <http://www.tuulivoimayhdistys.fi/tietoa-tuulivoimasta/tietoa-tuulivoimasta/tuulivoima-suomessa-ja-maailmalla/tuulivoima-suomessa> (Viitattu: 14. heinäkuuta 2017).

Zdonik, S. ja Katz, J. (2014) *Guide to OSI and TCP / IP Models*. Springer. Saatavissa: <http://library.books24x7.com/toc.aspx?bookid=76720>.

Hardware Setting & Mode Configuration

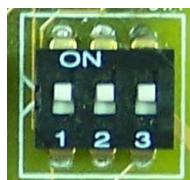
Jumper Settings for RS-422 or RS-485

Inside the unit, there is one 3-pin DIP switch which is set to select the mode of operation. You need to open the case then you can set the switch settings to RS-422 mode, or RS-485 mode, as per the requirements of your application. After setting of switches, you then proceed to insert the driver CD and start driver installation.



The RS-422 & RS-485 Mode Block Configuration Settings are listed as follows.

RS-422 & RS-485 Mode Block Configuration



SW (External DIP Switch) for Mode Setting

	Operation Mode	S1	S2	S3
RS-422	4 wire with Handshaking	ON	ON	ON
RS-485	Full Duplex (4 wire)	OFF	ON	ON
	Half Duplex (2 wire) with Echo	OFF	OFF	ON
	Half Duplex (2 wire) - without Echo	OFF	OFF	OFF

Example jumper block setting (RS-422 mode)



Inside the unit, there is one 3 x 7 (21 pin) header blocks which are jumpered to enable Tx, Rx, CTS 120 Ohm termination resistors and Tx, Rx 750 Ohm BIASing resistor.

You will need to open the case than set the jumper setting for RS-422 mode or RS-485 mode as per the requirements of your application.

Settings are listed as follows:

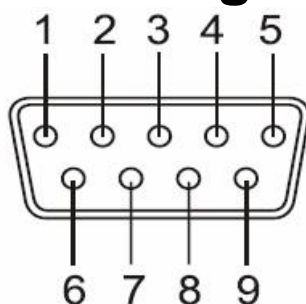
Jumper	Function
1-2 enable	Tx Termination of 120 Ohm. This jumper should always be
2-3 disable	populated for RS-485 mode.

4-5	enable	Pull-up Tx+ to VCC by 750 Ohm Bias resistor. This jumper should	
5-6	disable	be populated for pull-up Tx+.	
7-8	enable	Pull-down Tx- to GND by 750 Ohm Bias resistor. This jumper should	
8-9	disable	be populated for pull-down Tx- .	
10-11	enable	Rx Termination of 120 Ohm. This jumper should always be	
11-12	disable	populated for RS-422 mode.	
13-14	enable	Pull-up Rx+ to VCC by 750 Ohm Bias resistor. This jumper should	
14-15	disable	be populated for pull-up Rx+.	
16-17	enable	Pull-down Rx- to GND by 750 Ohm Bias resistor. This jumper should	
17-18	disable	be populated for pull-down Rx-.	
19-20	enable	CTS Termination of 120 Ohm. This jumper should always be	
20-21	disable	populated for RS-422 mode.	

Note : all other positions = no jumper populated.

Sometimes, when operating in RS-422 or RS-485, it is necessary to configure 120 Ohm termination of the data transmission lines. Generally this must be done in the cabling, since this depends on the installation of connections. Before applying the option, check your cable specification for proper impedance matching.

RS-422/485 Pin-outs & RS-422/485 Signal Wiring

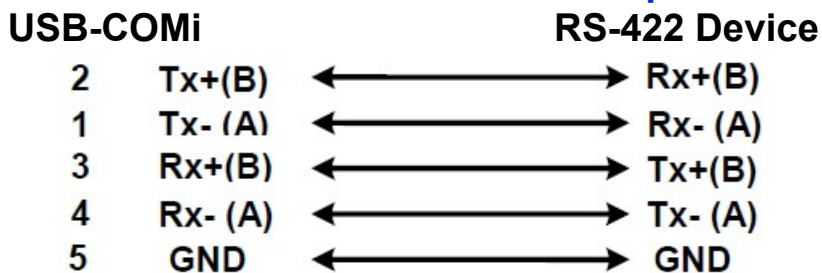


RS-422 Signal Pin-outs of DB-9 Male

Pin 1	TxD- (A)
Pin 2	TxD+(B)
Pin 3	RxD+(B)
Pin 4	RxD-(A)
Pin 5	GND
Pin 6	RTS- (A)
Pin 7	RTS+(B)
Pin 8	CTS+(B)
Pin 9	CTS- (A)

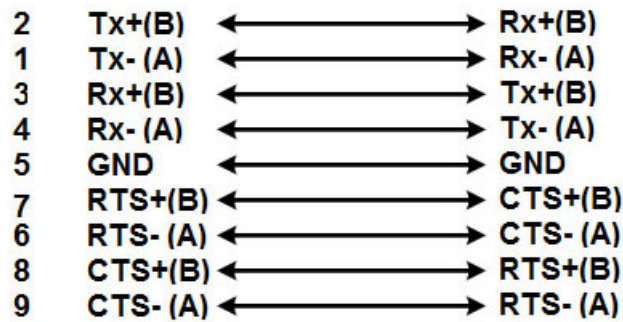
RS-422 Signal Wiring

● Point-to-Point 4 Wire Full Duplex



● RS-422 with Handshaking



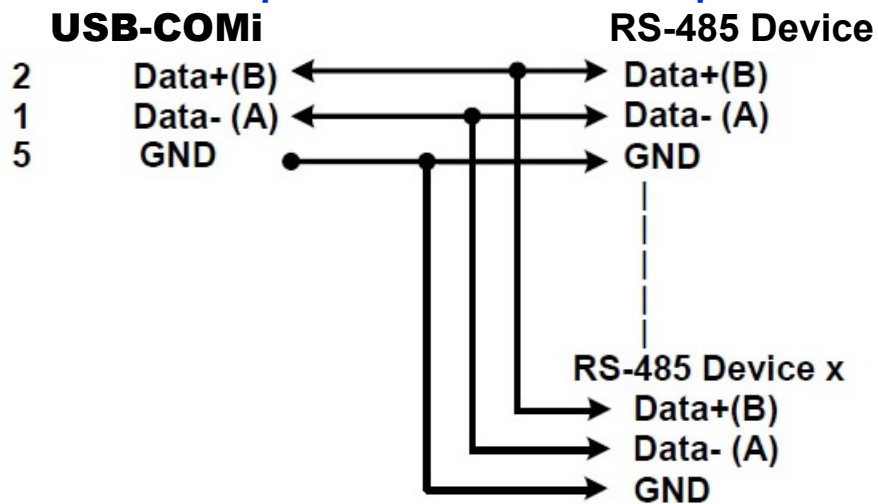


RS-485 2-Wire (Half duplex) Signal Pin-outs of DB-9 Male

Pin 1	Data- (A)
Pin 2	Data+(B)
Pin 5	GND

RS-485 Signal Wiring

● Multidrop RS-485 2-Wire Half-duple



LIITE B

Saatavissa: https://www.google.fi/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwi3nLPloqDXAhWLLIAKHW1mAi0QFgguMAE&url=http%3A%2F%2Fhanjindata.lgnas.com%3A10000%2Fmyweb%2FP1339%2FPelcoDProtocolManual_1_1.pdf&usg=AOvVaw2volD7FpOg7mp3oRLcGJh0

Standard Number	TF-0002				
Version	1	Revision	1	Release Date	July 7, 2003
Key words:	D protocol, serial, extended, Sync byte, 0xFF, Advanced Feature Set, Extended Response				



PELCO “D” PROTOCOL MANUAL

TABLE OF CONTENTS

DOCUMENT HISTORY.....	3
LEGAL NOTICES.....	4
NOTICE OF DISCLAIMER	4
PROPRIETARY NOTICE	4
QUESTIONS.....	4
WHAT THIS MANUAL COVERS	5
THE BYTE FORMAT.....	5
THE MESSAGE FORMAT.....	5
The Standard Command Set	6
Extended Commands	7
Advanced Feature Set	8
Creating Labels.....	9
EXAMPLE MESSAGES.....	9
Responses	10
Opcode Descriptions	11
APPENDIX A.....	17
ADVANCED FEATURE COMMANDS.....	17
INDEX.....	19

Document History

1. Initial Release, 7/07/03

LEGAL NOTICES

NOTICE OF DISCLAIMER

Pelco makes no claims, expressed or implied, regarding the usefulness of this protocol, it's implementation, or it's correctness. Any use of this protocol is the sole responsibility of the agency implementing the protocol. The contents of this document and the function of the protocol are subject to change without notice.

PROPRIETARY NOTICE

The contents of this document are considered to be the property of Pelco. Users of this protocol agree to use the protocol only in the interests of Pelco. Any use of this protocol to Pelco's detriment is prohibited.

Those receiving this protocol cannot redistribute the protocol without the expressed written consent of Pelco.

QUESTIONS

Questions regarding this protocol, it's implementation, use, and distribution should be addressed to:

Pelco
3500 Pelco Way
Clovis, California, USA 93612-5699

(559) 292-1981 Voice
(559) 292-1018 FAX

WHAT THIS MANUAL COVERS

This manual describes the minimum requirements for implementing the Pelco “D” protocol. This protocol is used to communicate between a controlling device (e.g. a matrix switching system) and a receiver/driver (e.g. a dome drive).

Not all devices will be able to accommodate all of the features available in this protocol. This protocol is designed to cover the feature sets of a wide variety of equipment.

THE BYTE FORMAT

Transmitters will format a single character and receivers will be able to decipher a single character as: 1 start bit, 8 data bits, 1 stop bit, and no parity.

THE MESSAGE FORMAT

The format for a message is:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Sync Byte	Address	Command 1	Command 2	Data 1	Data 2	Checksum

Note that values in this document prefixed with “0x” are hexadecimal numbers.

The *synchronization byte* (Sync Byte) is *always* **0xFF**.

The *Address* is the logical address of the receiver/driver device being controlled.

The *Checksum* is calculated by performing the 8 bit (modulo 256) sum of the payload bytes (bytes 2 through 6) in the message.

THE STANDARD COMMAND SET

Command 1 and 2 are represented as follows:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command 1	Sense	Reserved	Reserved	Auto / Manual Scan	Camera On / Off	Iris Close	Iris Open	Focus Near
Command 2	Focus Far	Zoom Wide	Zoom Tele	Down	Up	Left	Right	<i>Always 0</i>

A value of ‘1’ entered in the bit location for the function desired will enable that function. A value of ‘0’ entered in the same bit location will disable or ‘stop’ the function.

The sense bit (command 1 bit 7) indicates the meaning of bits 4 and 3. If the sense bit is on (value of ‘1’), and bits 4 and 3 are on, the command will enable auto-scan and turn the camera on. If the sense bit is off (value of ‘0’), and bits 4 and 3 are on the command will enable manual scan and turn the camera off. Of course, if either bit 4 or bit 3 are off then no action will be taken for those features.

The *reserved bits* (6 and 5) should be set to 0.

Byte 5 contains the pan speed. Pan speed is in the range of ‘0x00’ to ‘0x3F’ (high speed) and ‘0x40’ for “turbo” speed. Turbo speed is the maximum speed the device can obtain and is considered separately because it is not generally a smooth step from high speed to turbo. That is, going from one speed to the next usually looks smooth and will provide for smooth motion with the exception of going into and out of turbo speed. A pan speed value of ‘0x00’ results in very slow motion, not cessation of motion. To stop pan motion both the Left and Right direction bits must be turned off – set to ‘0’ – regardless of the value set in the pan speed byte.

Byte 6 contains the tilt speed. Tilt speed is in the range of ‘0x00’ to ‘0x3F’ (maximum speed). Turbo speed is not allowed for the tilt axis. A tilt speed value of ‘0x00’ results in very slow motion, not cessation of motion. To stop tilt motion both the Down and Up direction bits must be turned off – set to ‘0’ – regardless of the value set in the tilt speed byte.

Byte 7 is the checksum. The checksum is the 8 bit (modulo 256) sum of the payload bytes (bytes 2 through 6) in the message.

EXTENDED COMMANDS

In addition to the “PTZ” commands shown above, there are control commands that allow access to the more advanced features of some equipment. Bytes 3 and 4 can be thought of as the command’s opcodes.

Command	Byte 3	Byte 4	Byte 5	Byte 6	Response Type
Set Preset	00	0x03	00	Preset id	General
Clear Preset	00	0x05	00	Preset id	General
Go To Preset	00	0x07	00	Preset id	General
Flip (180° about)	00	0x07	00	0x21	General
Go To Zero Pan	00	0x07	00	0x22	General
Set Auxiliary	00	0x09	00	01 to 08	General
Clear Auxiliary	00	0x0B	00	01 to 08	General
Remote Reset	00	0x0F	00	00	General
Set Zone Start	00	0x11	00	01 to 08	General
Set Zone End	00	0x13	00	01 to 08	General
Write Character to Screen	00	0x15	Column 00 to 0x27	ASCII Value	General
Clear Screen	00	0x17	00	00	General
Alarm Acknowledge	00	0x19	00	01 to 08	General
Zone Scan On	00	0x1B	00	00	General
Zone Scan Off	00	0x1D	00	00	General
Set Pattern Start	00	0x1F	00	Pattern id	General
Set Pattern Stop	00	0x21	00	00	General
Run Pattern	00	0x23	00	Pattern id	General
Set Zoom Speed	00	0x25	00	00 to 03	General
Set Focus Speed	00	0x27	00	00 to 03	General
Reset Camera to defaults	00	0x29	00	00	General
Auto-focus auto/on/off	00	0x2B	00	00-02	General
Auto Iris auto/on/off	00	0x2D	00	00-02	General
AGC auto/on/off	00	0x2F	00	00-02	General
Backlight compensation on/off	00	0x31	00	01-02	General
Auto white balance on/off	00	0x33	00	01-02	General
Enable device phase delay mode	00	0x35	00	00	General
Set shutter speed	00	0x37	Any	Any	General
Adjust line lock phase delay	00-01	0x39	Any	Any	General

Command	Byte 3	Byte 4	Byte 5	Byte 6	Response Type
Adjust white balance (R-B)	00-01	0x3B	Any	Any	General
Adjust white balance (M-G)	00-01	0x3D	Any	Any	General
Adjust gain	00-01	0x3F	Any	Any	General
Adjust auto-iris level	00-01	0x41	Any	Any	General
Adjust auto-iris peak value	00-01	0x43	Any	Any	General
Query ¹	00	0x45	Any	Any	See “Responses” part of this document.

ADVANCED FEATURE SET

Command	Byte 3	Byte 4	Byte 5	Byte 6	Response Type
Reserved Opcode	00	0x47	00	00	Not Applicable
Set Zero Position	00	0x49	00	00	General
Set Pan Position	00	0x4B	Pan position MSB	Pan position LSB	General
Set Tilt Position	00	0x4D	Tilt position MSB	Tilt position LSB	General
Set Zoom Position	00	0x4F	Zoom position MSB	Zoom position LSB	General
Query Pan Position	00	0x51	00	00	Extended (0x59)
Query Tilt Position	00	0x53	00	00	Extended (0x5B)
Query Zoom Position	00	0x55	00	00	Extended (0x5D)
Reserved Opcode	00	0x57	00	00	Not Applicable
Query Pan Response	00	0x59	Pan position MSB	Pan position LSB	Not Applicable
Query Tilt Response	00	0x5B	Tilt position MSB	Tilt position LSB	Not Applicable
Query Zoom Response	00	0x5D	Zoom position MSB	Zoom position LSB	Not Applicable
Set Magnification	00	0x5F	Mag position MSB	Mag position LSB	General
Query Magnification	00	0x61	00	00	Extended (0x63)
Query Magnification Response	00	0x63	Mag position MSB	Mag position LSB	Not Applicable
Reserved Opcode	00	0x65	00	00	Not Applicable

¹ This command can only be used in a point to point application. A device being queried will respond to any address. If more than one device hears this command, multiple devices will transmit at the same time.

Command	Byte 3	Byte 4	Byte 5	Byte 6	Response Type
Reserved Opcode	00	0x67	00	00	Not Applicable
Reserved Opcode	00	0x69	00	00	Not Applicable
Reserved Opcode	00	0x6B	00	00	Not Applicable
Reserved Opcode	00	0x6D	00	00	Not Applicable
Reserved Opcode	00	0x6F	00	00	Not Applicable
Reserved Opcode	00	0x71	00	00	Not Applicable

CREATING LABELS

Many devices have the ability to display labels on the video. Labels that identify the preset or zone being scanned are common. There is a special technique to establish a label that is associated with either a preset or a zone. First, send the label to the receiver/driver using the “Write Character to Screen” command. After the label is on the screen, set the preset or zone. That will establish the label and associate it with the preset. Reference the detailed information in the “Opcode Descriptions” section of this manual.

EXAMPLE MESSAGES

Message to send	Message
Receiver 1, Camera on	0xFF, 0x01, 0x88, 0x00, 0x00, 0x00, 0x89
Receiver 1, Camera off	0xFF, 0x01, 0x08, 0x00, 0x00, 0x00, 0x09
Receiver 2, Pan Left	0xFF, 0x02, 0x00, 0x04, 0x20, 0x00, 0x26
Receiver 2, Stop	0xFF, 0x02, 0x00, 0x00, 0x20, 0x00, 0x22
Receiver 10, Camera on, Focus far, Tilt Down	0xFF, 0x0A, 0x88, 0x90, 0x20, 0x00, 0x42

Note: the checksum calculation for the last message looks like this:

0xFF	1111 1111	Sync byte is <i>not used</i> for the checksum
0x0A	0000 1010	
0x88	<u>1000 1000</u>	
Subtotal	1001 0010	0x92
0x90	<u>1001 0000</u>	
Subtotal	0010 0010	0x22 (modulo 256 allows the high bit to roll off)
0x20	<u>0010 0000</u>	
Subtotal	0100 0010	0x42
0x00	<u>0000 0000</u>	
	0100 0010	0x42 Final checksum value

RESPONSES

Devices that receive a “D” protocol command may generate a response. The response formats are described below.

The General Response

The General Response has the following format. Note that each block represents 1 byte.

Byte 1	Byte 2	Byte 3	Byte 4
Sync	Address	Alarm Information	Checksum

The alarm information is formatted as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
None	Alarm 7	Alarm 6	Alarm 5	Alarm 4	Alarm 3	Alarm 2	Alarm 1

If the bit is on (1) then the alarm is active. If the bit is off (0) then the alarm is inactive.

The checksum is the sum of the *transmitted command’s checksum* and the alarm information.

The Query (0x45) Response

The response to the Query command is:

Byte 1	Byte 2	Bytes 3 to 17	Byte 18
Sync (1 byte)	Address (1 byte)	Part Number (15 bytes)	Checksum (1 byte)

The address is the address of the device responding to the query. The part number is the ASCII text string containing the program number of the device being queried.

The checksum is the 8 bit (modulo 256) sum of the *transmitted query command’s checksum*, the address of the response, and the 15-byte part number.

The Extended Response

The Extended Response has the following format. Note that each block represents 1 byte

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Sync	Address	Future Use	“opcode”	Data1	Data2	Checksum

The address is the address of the device that is responding.

The Future Use byte should always be set to 0.

Opcode, Data1 and Data2 are dependent on the type of response. See the opcode description section of this document for the details of a particular response.

The checksum is the 8 bit (modulo 256) sum of all the bytes excluding the Sync byte.

OPCODE DESCRIPTIONS

Please note that this is not an exhaustive list, the most commonly used commands have been described for clarity of implementation.

Set Preset (0x03)

Clear Preset (0x05)

Go To Preset (0x07)

The parameter in byte 6 of these commands is the ID of the preset to be acted on. Valid preset IDs begin at 1. Most devices support at least 32 presets. Refer to the manual of the device under use for information about what range of presets are valid for that equipment.

Write Character To Screen (0x15)

The parameter in **byte 5** of this command indicates the column to write to. This parameter is interpreted as follows:

Columns 0-19 are used to receive **zone** labels. ²

Columns 20-39 are used to receive **preset** labels. ³

² For Spectra III and Spectra III SE *only*, characters written to these positions are not written directly to the screen. Once the *SET_ZONE_START* (opcode 0x11) command is received, the characters are displayed.

³ For Spectra III and Spectra III SE *only*, characters written to these positions are not written directly to the screen. However, if characters are written to these columns and no *SET_PRESET* (opcode 0x03) command is received within *250 milliseconds*, the characters will be displayed on the screen beginning at the first column of the second row of the display.

Set Pattern Start (0x1F)

Run Pattern (0x23)

The parameter in byte 6 of these commands indicates the pattern to be set/run. *Spectra III* and *Spectra III SE* interpret this byte as follows:

Value	Action
0 or 1	Sets/runs pattern 1
2	Sets/runs pattern 2
3	Sets/runs pattern 3
4	Sets/runs pattern 4

Spectra II and Esprit interpret this byte as follows:

Value	Action
0	Sets/runs the single “long pattern”
1	Sets/runs the first “short pattern”
2	Sets/runs the second “short pattern”

Reserved Opcode (0x47)

For Pelco internal use only.

Set Zero Position (0x49)

This command is used to set the pan position that the unit uses as a zero reference point for the azimuth on-screen display. The unit's current pan position when this command is received becomes the zero reference point. This command performs the same function as the "Set Azimuth Zero" menu item.

Set Pan Position (0x4B)

This command is used to set the pan position of the device. The position is given in hundredths of a degree and has a range from 0 to 35999 (decimal). Example: the value to use to set the pan position to 45 degrees is 4500. Note that the value used here is always the "absolute" pan position. It **does not** take into account any adjustment to the screen display that may have been made by using the "Set Zero Position", opcode (0x49) command or the "Set Azimuth Zero" menu item.

Set Tilt Position (0x4D)

This command is used to set the tilt position of the device. The position is given in hundredths of a degree and has a range from 0 to 35999 (decimal). Generally these values are interpreted as follows: *Zero degrees* indicates that the device is pointed horizontally (at the *horizon*). *Ninety degrees* indicates that the device is pointed straight down.

Examples:

- 1) the value used to set the tilt position to 45 degrees *below the horizon*, is 4500.
- 2) the value used to set the tilt position 30 degrees *above the horizon*, is 33000.

Note that different equipment will have different ranges of motion. To determine the abilities of a specific piece of equipment, refer to that device's operation manual.

Set Zoom Position (0x4F)

This command is used to set the zoom position of the device. The position is given as a ratio based on the device's Zoom Limit setting. The position is calculated as follows:

$$\text{Position} = (\text{desired_zoom_position} / \text{zoom_limit}) * 65535$$

Where desired_zoom_position and zoom_limit are given in units of magnification.

Example: Given that the zoom limit of the device's camera is X184, calculate the value needed to set the zoom position to X5:

$$\text{Position} = (5 / 184) * 65535 = \text{approximately } 1781$$

Query Pan Position (0x51)

This command is used to query the current pan position of the device. The response to this command uses opcode 0x59. See the description of opcode 0x59 for more information.

Query Tilt Position (0x53)

This command is used to query the current tilt position of the device. The response to this command uses opcode 0x5B. See the description of opcode 0x5B for more information.

Query Zoom Position (0x55)

This command is used to query the current zoom position of the device. The response to this command uses opcode 0x5D. See the description of opcode 0x5D for more information.

Reserved Opcode (0x57)

For Pelco internal use only.

Query Pan Position Response (0x59)

The position is given in hundredths of a degree and has a range from 0 to 35999 (decimal).

Example: a position value of 4500 indicates 45 degrees. Note that the value returned is always the “absolute” pan position. It **does not** take into account any adjustment to the screen display that may have been made by using the “Set Zero Position”, opcode (0x49) command or the “Set Azimuth Zero” menu item.

Query Tilt Position Response (0x5B)

The position is given in hundredths of a degree and has a range from 0 to 35999 (decimal). Refer to examples listed in description of the “Set Tilt Position”, opcode 0x4D command.

Query Zoom Position Response (0x5D)

The position is given as a ratio based on the device’s Zoom Limit setting. This value can be converted into units of magnification by using the following formula:

$\text{current_magnification} = (\text{position} / 65535) * \text{zoom_limit}$

Where `current_zoom_position` and `zoom_limit` are given in units of magnification.

Example: Given that the zoom limit of the device’s camera is X184, position value is 1781, calculate the current magnification:

Current magnification = $(1781 / 65535) * 184 =$ approximately X5.

Note: This message is sent in response to the Query Zoom Position (0x55) command.

Set Magnification (0x5F)

This command is used to set the zoom position of the device. The position is given in hundredths of units of magnification. Example: a value of 500 means X5.

Query Magnification (0x61)

This command is used to query the current zoom position of the device. The response to this command uses opcode 0x63. See the description of opcode 0x63 for more information.

Query Magnification Response (0x63)

The value returned is given in hundredths of units of magnification. Example: a value of 500 means X5.

Reserved Opcode (0x65)

For Pelco internal use only.

Reserved Opcode (0x67)

For Pelco internal use only.

Reserved Opcode (0x69)

For Pelco internal use only.

Reserved Opcode (0x6B)

For Pelco internal use only.

Reserved Opcode (0x6D)

For Pelco internal use only.

Reserved Opcode (0x6F)

For Pelco internal use only.

Reserved Opcode (0x71)

For Pelco internal use only.

Appendix A

Advanced Feature Commands Equipment Compatibility⁴

	Spectra III, Spectra III SE v1.20 and later	Esprit ES3xC, ES3xPC v3.05 and later
Reserved Opcode (0x47)	No	No
Set Zero Position (0x49)	Yes	Yes
Set Pan Position (0x4B)	Yes	Yes
Set Tilt Position (0x4D)	Yes	Yes
Set Zoom Position (0x4F)	Yes	No
Query Pan Position (0x51)	Yes	Yes
Query Tilt Position (0x53)	Yes	Yes
Query Zoom Position (0x55)	Yes	No
Reserved Opcode (0x57)	No	No
Query Pan Response (0x59)	Yes	Yes
Query Tilt Response (0x5B)	Yes	Yes
Query Zoom Response (0x5D)	Yes	No
Set Magnification (0x5F)	Yes	Yes
Query Magnification (0x61)	Yes	Yes

⁴ Pelco products not listed in this table do not implement the ‘Advanced Feature’ command set and will not be modified to accommodate these features in the future.

Query Magnification Response (0x63)	Yes	Yes
Reserved Opcode (0x65)	No	No
Reserved Opcode (0x67)	No	No
Reserved Opcode (0x69)	No	No
Reserved Opcode (0x6B)	No	No
Reserved Opcode (0x6D)	No	No
Reserved Opcode (0x6F)	No	No
Reserved Opcode (0x71)	No	No

INDEX

A

Address	4, 5
ADVANCED FEATURE SET	8
Appendix A	17
Autoscan	6

B

Byte 1	5
Byte 2	5
Byte 3	5
Byte 4	5
Byte 5	5, 6
Byte 6	5
Byte 7	5, 6

C

Camera off	6
Camera on	6
Checksum	5, 6, 9
Command 1	5, 6
Command 2	5, 6
CREATING LABELS	9

D

Data 1	5
Data 2	5
Document History	3

E

Example Messages	9
EXTENDED COMMANDS	7

L

Legal Notices	4
---------------------	---

M

Manual scan	6
-------------------	---

N

Notice of Disclaimer	4
----------------------------	---

O

OPCODE DESCRIPTIONS	11
---------------------------	----

P

Pan speed	6
Phone	4
Proprietary Notice	4

Q

Questions	4
-----------------	---

R

RESPONSES	10
-----------------	----

S

Sense bit	6
Synchronization	5

T

THE BYTE FORMAT	5
THE MESSAGE FORMAT	5
THE STANDARD COMMAND SET	6
Tilt speed	6
Turbo speed	6

W

WHAT THIS MANUAL COVERS	5
-------------------------------	---

5 Application programming interface

Introduction

This chapter described the communication protocol between the Robin drone detection system and any system that requires real-time track information. Examples of these systems are drone defence systems, like spoofing systems or interceptors. It can also be used to feed track information into a command and control system or visualization system.

The protocol follows a streaming pattern; no two-way communication is required. It can therefore be used on any communication channel that supports either peer-to-peer communication or broadcast communication.

The communication protocol is based on XML, which is one of the most versatile formats for exchanging information. The contents can be very well defined by using an XML Schema Definition (XSD).

The implementation on Elvira Search is through a TCP server connection. This server listens on port 16810 (configurable) and streams the XML. Each update sends one root element. Each closing tag is followed by a newline character to facilitate parsing. The server supports only one connection.

Schema layout

The root element is named "Robin". The next level element is one or more of the following elements:

- *Status*
- *Tracks*
- *Events*

The *status* element is used to convey status information such as the operational state of the system, any warning or errors that may have been detected at the system level.

The *tracks* element is used to send current information about the radar tracks. Each track has a unique identifier so changes over time can be observed. The latest state of each track is represented, even if it has not changed since the last update. A timestamp of the last update is included in the state. Tracks that are no longer in the list can be considered to be gone / dropped.

The *event* element can be used to actively warn client systems about certain events. An example could be an alarm if a drone has been detected. Another example could be a warning about degraded performance due to interference. These events are configurable and most likely application specific. For the purpose of this document we consider this element a placeholder for future expansion.

Status

A Robin radar system consists of a central processing system and one or more sensors. These sensors can be of different types. The status element contains status information about all of these sensors.

The *SystemStatus* element contains the following elements:

1. *Name*; a generic name of the system
2. *Version*; a version number of the software
3. *OperationalState*; the operational state of the central processing system, i.e. *idle*, *operational*, *initializing*, *error*
4. *Messages*; contains any number of child elements for event messages. These can be any type of messages, but most commonly used for error messages.
5. *Sensors*; a list of *SensorStatus* elements for each sensor system attached.

The *SensorStatus* element contains the following elements:

1. *Name*; a generic name of the system

2. *Version*; a version number of the software
3. *Position*; location (latitude, longitude, altitude) of the device using the WGS84 coordinate system. Latitude and longitude are in decimal degrees, altitude in meters above mean sea level.
4. *Type*: the type of sensor system, i.e. *search radar*, *acquisition radar*, *camera*, etc.
5. *OperationalState*; i.e. *idle*, *operational*, *initializing*, *error*
6. *Messages*; contains any number of child elements for event messages. These can be any type of messages, but most commonly used for error messages.

4. Tracks

The *Tracks* element contains the following elements:

1. A sequence of *Track* elements
2. *<attribute> Timestamp*; the time this track list was produced, i.e. all track information is up to date up to this timestamp.

The *Track* element contains the following elements:

1. *Timestamp*; the time this track was last updated.
2. *Position*; current location (latitude, longitude, altitude) of the track using the WGS84 coordinate system. Latitude and longitude are in decimal degrees, altitude in meters above mean sea level.
3. *Velocity*; the speed vector of the track. Defined as bearing in degrees, elevation (change of altitude) in degrees and velocity in meters/second.
4. *Classification*; The classification of the track, i.e. Bird, Vehicle, Aircraft.
5. *Reflection*; the measured radar reflection of the track which is indicative for the radar cross section (RCS). The unit is dB and defined as the measured signal-to-noise ratio of the track corrected for range.
6. *Score*; the score of the track, a measure of the tracking quality. Value is between 0 and 1, where 0 is poor and 1 is a perfect quality track.

Examples

Status message

```
<Robin>
  <SystemStatus>
    <Name>Sys1</Name>
    <Version>1.2.3.a</Version>
    <OperationalState>idle</OperationalState>
    <Messages>
      <Message>message content</Message>
    </Messages>
    <Sensors>
      <SensorStatus>
        <Name>Sens1</Name>
        <Version>1.2.3.b</Version>
        <OperationalState>idle</OperationalState>
        <Position>
          <Latitude>52.456</Latitude>
          <Longitude>4.789</Longitude>
          <Altitude>34</Altitude>
        </Position>
        <SensorType>SearchRadar</SensorType>
        <Messages/>
      </SensorStatus>
    </Sensors>
  </SystemStatus>
</Robin>
```

Track update message

```
<Robin>
  <Tracks TimeStamp="2002-05-30T09:30:10.5">
    <Track id="678">
      <TimeStamp>2002-05-30T09:30:10.5</TimeStamp>
      <Position>
        <Latitude>52.456</Latitude>
        <Longitude>4.789</Longitude>
        <Altitude>34</Altitude>
      </Position>
      <Velocity>
        <Azimuth>34.6</Azimuth>
        <Elevation>1.45</Elevation>
        <Speed>18.9</Speed>
      </Velocity>
      <Classification id="4">Large Bird</Classification>
      <Reflection>45.6</Reflection>
      <Score>0.56</Score>
    </Track>
    <Track id="679">
      <TimeStamp>2002-05-30T09:31:15.5</TimeStamp>
      <Position>
        <Latitude>53.456</Latitude>
        <Longitude>3.789</Longitude>
        <Altitude>35</Altitude>
      </Position>
      <Velocity>
        <Azimuth>14.6</Azimuth>
        <Elevation>2.45</Elevation>
        <Speed>18.9</Speed>
      </Velocity>
      <Classification id="2">Vehicle</Classification>
      <Reflection>85.6</Reflection>
      <Score>0.99</Score>
    </Track>
  </Tracks>
</Robin>
```


LIITE D: ASETTUMISTARKKUUDEN MITTAUSTULOKSET

Sivusuunta

H = Home, R = Right/positiivinen kulma, L = Left/ negatiivinen kulma

Kulmamittauksen 0 = Home Position

Komento	Aika	Mitattu suunta	Nopeus ast/s	Asettumisvirhe Edellinen vs. seuraava	Asettumisvirhe Ensimmäinen vs. nykyinen	
H		0,0				
R 2		87,5	43,8			
L 4		-93,0	45,1			
R 4		84,0	44,3	3,5	3,5	
L 4		-97,5	45,4	4,5	4,5	
R 4		80,0	44,4	4,0	7,5	
L 4		-101,5	45,4	4,0	8,5	
R 4		76,0	44,4	4,0	11,5	
L 4		-105,5	45,4	4,0	12,5	
R 4		72,0	44,4	4,0	15,5	
L 4		-109,5	45,4	4,0	16,5	
R 4		68,0	44,4	4,0	19,5	
H		0,5		4,0	<--ka	
Mittausjärjestelyn nollaus						
H		0,0				
R 2		89,0	44,5			
L 4		-93,5	45,6			
R 4		85,0	44,6	4,0	4,0	
L 4		-97,0	45,5	3,5	3,5	
R 4		81,0	44,5	4,0	8,0	
L 4		-101,0	45,5	4,0	7,5	
R 4		78,0	44,8	3,0	11,0	
L 4		-105,0	45,8	4,0	11,5	
R 4		74,0	44,8	4,0	15,0	
L 4		-108,0	45,5	3,0	14,5	
R 4		71,0	44,8	3,0	18,0	
H		0,0		3,6	10,3	<--ka
R 2		89,0	44,5		0,0	
H		0,0		0,0	0,0	
R 2		89,0	44,5	0,0	0,0	
H		0,0		0,0	0,0	
L 2		-92,0	46,0			
H		0,5		0,5	0,5	
L 2		-91,0	45,8	1,0	1,0	
H		0,5		0,0	0,5	
R 2		90,0	44,8	1,0	1,0	
H		0,0		0,5	0,0	
Mittausjärjestelyn nollaus				0,4	0,3	<--ka

H	0,0			
R 1	44,0	44,0		
R 1	89,0	45,0		
R 1	133,0	44,0		
H	0,0		0,0	0,0
R 3	134,0	44,7	1,0	1,0
H	0,0		0,0	0,0
L 1	-46,0	46,0		
L 1	-92,0	46,0		
L 1	-138,0	46,0		
H	0,0		0,0	0,0
L 3	-137,0	45,7	1,0	1,0
H	0,0		0,0	0,0
	ka	45,0	0,3	0,3

Korosuunta

H = Home, U = Up/ positiivinen kulma, D = Down/ negatiivinen kulma

Kulmamittauksen 0 = vaakataivo

Komento	Aika	Mitattu suunta	Nopeus ast/s	Asettumisvirhe Edellinen vs. seuraava	Asettumisvirhe Ensimmäinen vs. nykyinen	
H		-4,5				
D 3		-50,5	15,3			
U 6		42	15,4			
D 6		-46	14,7	4,5	4,5	
U 6		43	14,8	1,0	1,0	
D 6		-49	15,3	3,0	1,5	
U 6		43,5	15,4	0,5	1,5	
D 6		-48,5	15,3	0,5	2,0	
U 6		44	15,4	0,5	2,0	
D 6		-48	15,3	0,5	2,5	
U 6		45	15,5	1,0	3,0	
D 6		-47,5	15,4	0,5	3,0	
H		-4,5		1,3	<--ka	
Mittausjärjestelyn nollaus						
H		-4,5				
D 3		-51	15,5			
U 6		42	15,5			
D 6		-50	15,3	1,0	1,0	
U 6		43	15,5	1,0	1,0	
D 6		-49	15,3	1,0	2,0	
U 6		44	15,5	1,0	2,0	
D 6		-48	15,3	1,0	3,0	
U 6		45	15,5	1,0	3,0	
D 6		-47,5	15,4	0,5	3,5	
U 6		45,5	15,5	0,5	3,5	
D 6		-47	15,4	0,5	4,0	
H		-4,5		0,8	2,6	<--ka
D 3		-50,5	15,3		0,5	
H		-4,5		0,0	0,0	
D 3		-51	15,5	0,5	0,0	
H		-4,5		0,0	0,0	
U 3		41,5	15,3			
H		-5		0,5	0,5	
U 3		41,5	15,5	0,0	0,0	
H		-5		0,0	0,5	
D 3		-51	15,3	0,0	0,0	
H		-4,5		0,5	0,0	
Mittausjärjestelyn nollaus				0,2	0,2	<--ka

H	-4,5			
D 1	-20	15,5		
D 1	-35,5	15,5		
D 1	-50,5	15,0		
H	-4,5		0,0	0,0
D 3	-50,5	15,3	0,0	0,0
H	-4,5		0,0	0,0
U 1	10,5	15,0		
U 1	26	15,5		
U 1	42	16,0		
H	-4,5		0,0	0,0
U 3	42	15,5	0,0	0,0
H	-4,5		0,0	0,0
	ka	15,4	0,0	0,0

PAN AND TILT HEADS

PT-1020 Medium Duty, IP68, Lash <0,20°



Medium Duty Series

The PT-1020 is available in three different versions:

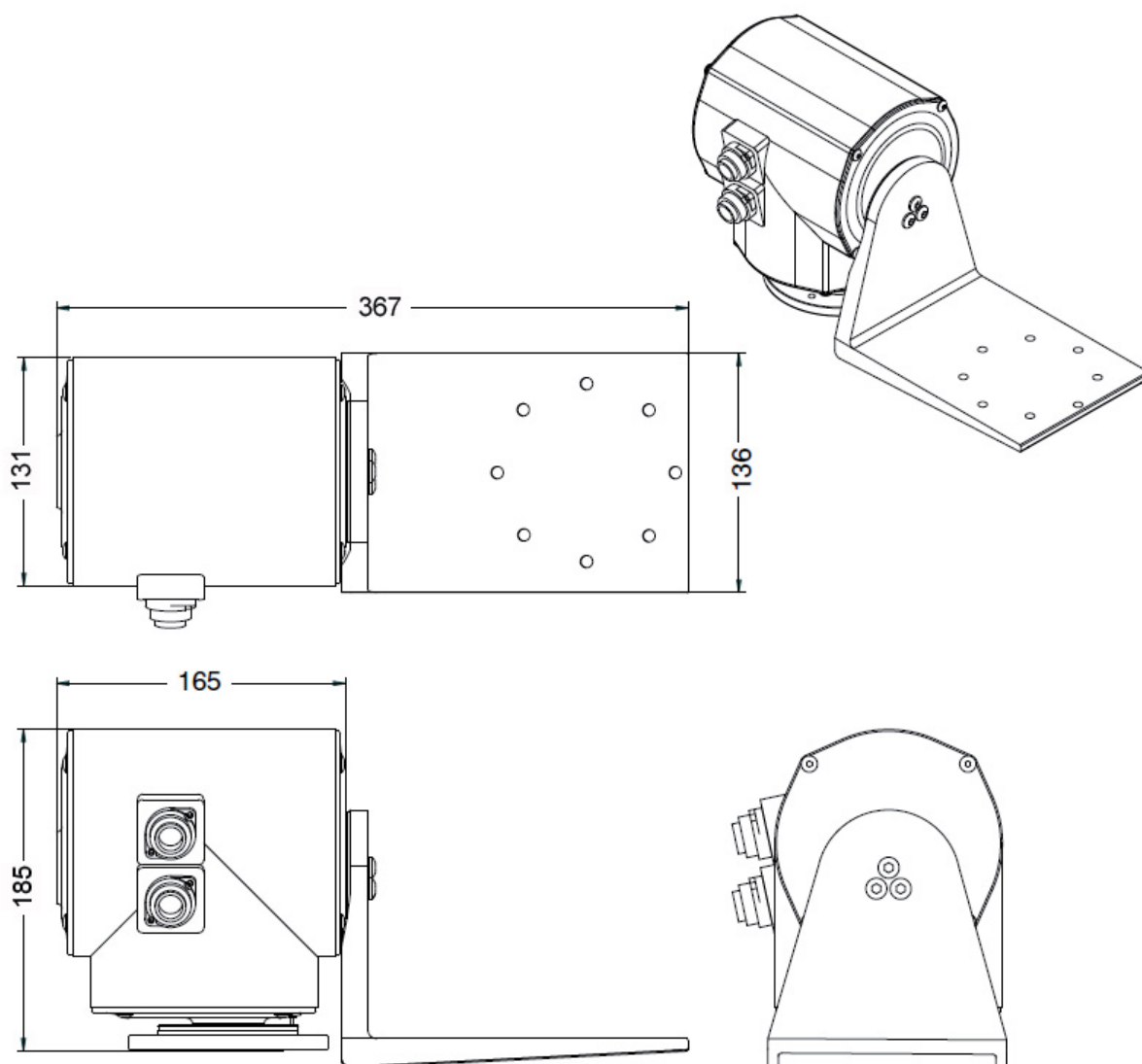
- Standard Version without receiver DC24V, DC12V, AC24V, AC110V, AC240V
- Stainless Steel 316 Version DC24V only. Top mount only
- Pelco D or Pelco P Version. RS-485 Receiver built-in, DC24V only

The receiver is full function with presets, tours, electronic endstops.

Max Carrying Capacity	Side mount format 25 kg	
	Top mount format 16 kg	
Voltages	DC24V, DC12V, AC24V, AC110V, AC240V	
Mechanical Size (mm)	Side mount 185(H)x367(W)x131(D) Top mount 202(H)x184(W)x131(D)	
Weight (kg)	< 6	
Speed (°/sec) AC	Pan: 3 or 6 or 12	Tilt: 3 or 6
Speed (°/sec) DC	Pan: 12 or 18 or 24 or 32 or 48	Tilt: 12 or 18
Operation Angle: Side Mount	Pan: 350°	Tilt: ± 180°
Operation Angle: Over the top Mount	Pan: 350°	Tilt: -90 ~ +25°
Temperature range	-25°C ~ +55°C Optional heater	
Color	Gray	
Lash	< 0.20°	
Protection	IP68	

PAN AND TILT HEADS

PT-1020 Medium Duty, IP68, Backlash <0,15°



All specifications and product descriptions are subject to changes without notice.

JHS 154 ETRS89-järjestelmään liittyvät karttaprojektiot, tasokoordinaatistot ja karttalehtijako, Liite 1: Projektiokaavat

Versio: 6.6.2008

Julkaistu:

Voimassaoloaika: Toistaiseksi

Transverse Mercator-projektiolle on esitetty laskentakaavoja esimerkiksi seuraavissa lähteissä:

- Hirvonen, R.A., Matemaattinen Geodesia, Teknillisen korkeakoulun ylioppilaskunta, Helsinki, 1972
- Hirvonen, R.A. The Use of Subroutines in Geodetic Computations, Maanmittaus, 1970.
- Hooijberg, Marten: Practical Geodesy, Springer Verlag Berlin Heidelberg New York 1997, pages 81-84
- Krüger L: Konforme Abbildung des Erdellipsoids in der Ebene, B. G. Teubner Verlag Leipzig 1912, pages 11-22
- König, R.; Weise, K. H.: Mathematische Grundlagen der höheren Geodäsie und Kartographie, Springer Verlag Berlin Göttingen Heidelberg, 1951
- Meade, B. K., Program for Computing Universal Transverse Mercator (UTM) Coordinates for Latitudes North or South and Longitudes East or West, Surveying and Mapping, Vol. 47, No. 1, pages 37 – 49.
- Poder, K.; Engsager, K.: Some Conformal Mappings and Transformations for Geodesy and Topographic Cartography, Publications 4 series, volume 6, National Survey and Cadastre Denmark, 1998

Seuraavaksi esitettävät kaavat ovat R.A. Hirvosen kirjasta Matemaattinen Geodesia ja Maanmittaus-lehden artikkelista.

Suomessa käytettävien ellipsoidien parametreja

Ellipsoidi	Isoakselin puolikas a (m)	Pikkuakselin puolikas b (m)	Litistyssuhde f
Kansainvälinen 1924 (Hayford 1909)	6378388.0	6356911.946128	1/297.0
GRS80	6378137.0	6356752.314140	1/298.257222101
WGS84	6378137.0	6356752.314245	1/298.257223563

Kaavoissa esiintyvät symbolit ja niiden määritelmät

kaavoissa käytetään kulmayksikkönä radiaania.

Symboli	Määritelmä
a	= ellipsoidin isoakselin puolikas
b	= ellipsoidin pikkuakselin puolikas
f	= ellipsoidin litistyssuhde
k_0	= mittakaavakerroin keskimeridiaanilla
λ_0	= projektion keskimeridiaani
E_0	= Itäkoordinaatin arvo keskimeridiaanilla
φ	= geodeettinen leveys
λ	= geodeettinen pituus
E	= projektion itäkoordinaatti

Symboli	Määritelmä
N	= projektion pohjoiskoordinaatti
γ	= meridiaanikonvergenssi
k	= mittakaavakerroin
A_1	= meridiaanin pituisen ympyrän säde
e^2	= ensimmäisen epäkeskisyyden neliö
e'^2	= toisen epäkeskisyyden neliö
n	= toinen litistysuhde
t	= suuntakulma tasolla (suuntakorjauksen kaavassa $\delta = T - t$)
c	= napakaarevuussäde
M	= meridiaanikaarevuussäde
N	= poikittaiskaarevuussäde (kaavassa (42))

Hyperboliset ja käänteiset hyperboliset (area) funktiot:

(Vain hyperbolisten funktioiden kaavoissa e = Neperin luku, muutoin e = ensimmäinen epäkeskisyyden)

$$\begin{aligned}\sinh(x) &= \frac{e^x - e^{-x}}{2} \\ \cosh(x) &= \frac{e^x + e^{-x}}{2} \\ \tanh(x) &= \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \operatorname{arsinh}(x) &= \ln(x + \sqrt{x^2 + 1}) \\ \operatorname{artanh}(x) &= \frac{1}{2} \ln\left(\frac{1+x}{1-x}\right) \\ \operatorname{sech}(x) &= \frac{1}{\cosh(x)} = \frac{2}{e^x + e^{-x}} \\ \operatorname{arsech}(x) &= \ln\left(\frac{1 + \sqrt{1 - x^2}}{x}\right)\end{aligned}$$

Apusuureet

$$n = \frac{f}{2 - f} = \frac{a - b}{a + b} \quad (01)$$

$$A_1 = \frac{a}{1 + n} \left(1 + \frac{n^2}{4} + \frac{n^4}{64}\right) \quad (02)$$

$$e^2 = 2f - f^2 \quad (03)$$

$$e'^2 = \frac{e^2}{1 - e^2} \quad (04)$$

$$V^2 = 1 + e'^2 \cos^2(\varphi) \quad (05)$$

$$h_1 = \frac{1}{2}n - \frac{2}{3}n^2 + \frac{37}{96}n^3 - \frac{1}{360}n^4 \quad (06)$$

$$\begin{aligned}
 h_2 &= \frac{1}{48}n^2 + \frac{1}{15}n^3 - \frac{437}{1440}n^4 \\
 h_3 &= \frac{17}{480}n^3 - \frac{37}{840}n^4 \\
 h_4 &= \frac{4397}{161280}n^4 \\
 h_1' &= \frac{1}{2}n - \frac{2}{3}n^2 + \frac{5}{16}n^3 + \frac{41}{180}n^4 \\
 h_2' &= \frac{13}{48}n^2 - \frac{3}{5}n^3 + \frac{557}{1440}n^4 \\
 h_3' &= \frac{61}{240}n^3 - \frac{103}{140}n^4 \\
 h_4' &= \frac{49561}{161280}n^4
 \end{aligned} \tag{07}$$

Projektiokaavat

Geodeettisista koordinaateista (φ, λ) tasokoordinaateiksi (N, E)

Syöte: pisteen geodeettiset koordinaatit (φ, λ)
Tulos: pisteen tasokoordinaatit (N, E) projektiotasolla,

$$Q' = \operatorname{arsinh}[\tan(\varphi)] \tag{08}$$

$$Q'' = \operatorname{artanh}[e \cdot \sin(\varphi)] \tag{09}$$

$$Q = Q' - e \cdot Q'' \tag{10}$$

$$l = \lambda - \lambda_0 \tag{11}$$

$$\beta = \arctan[\sinh(Q)] \tag{12}$$

$$\eta' = \operatorname{artanh}[\cos(\beta) \cdot \sin(l)] \tag{13}$$

$$\xi' = \arcsin\left[\frac{\sin(\beta)}{\operatorname{sech}(\eta')}\right] \tag{14}$$

$$\xi_1 = h_1' \sin(2\xi') \cosh(2\eta') \tag{15}$$

$$\xi_2 = h_2' \sin(4\xi') \cosh(4\eta')$$

$$\xi_3 = h_3' \sin(6\xi') \cosh(6\eta')$$

$$\xi_4 = h_4' \sin(8\xi') \cosh(8\eta')$$

$$\eta_1 = h_1' \cos(2\xi') \sinh(2\eta') \tag{16}$$

$$\eta_2 = h_2' \cos(4\xi') \sinh(4\eta')$$

$$\eta_3 = h_3' \cos(6\xi') \sinh(6\eta')$$

$$\eta_4 = h_4' \cos(8\xi') \sinh(8\eta')$$

$$\xi = \xi' + (\xi_1 + \xi_2 + \xi_3 + \xi_4) \tag{17}$$

$$\eta = \eta' + (\eta_1 + \eta_2 + \eta_3 + \eta_4) \tag{18}$$

$$N = A_1 \cdot \xi \cdot k_0 \tag{19}$$

$$E = A_1 \cdot \eta \cdot k_0 + E_0 \tag{20}$$

Tasokoordinaateista (N, E) geodeettisiin koordinaatteihin (φ, λ)

Syöte: pisteen tasokoordinaatit (N, E) projektiotasolla,
Tulos: pisteen geodeettiset koordinaatit (φ, λ)

$$\xi = \frac{N}{A_1 \cdot k_0} \quad (21)$$

$$\eta = \frac{E - E_0}{A_1 \cdot k_0} \quad (22)$$

$$\xi_1' = h_1 \sin(2\xi) \cosh(2\eta) \quad (23)$$

$$\xi_2' = h_2 \sin(4\xi) \cosh(4\eta)$$

$$\xi_3' = h_3 \sin(6\xi) \cosh(6\eta)$$

$$\xi_4' = h_4 \sin(8\xi) \cosh(8\eta)$$

$$\eta_1' = h_1 \cos(2\xi) \sinh(2\eta) \quad (24)$$

$$\eta_2' = h_2 \cos(4\xi) \sinh(4\eta)$$

$$\eta_3' = h_3 \cos(6\xi) \sinh(6\eta)$$

$$\eta_4' = h_4 \cos(8\xi) \sinh(8\eta)$$

$$\xi' = \xi - (\xi_1' + \xi_2' + \xi_3' + \xi_4') \quad (25)$$

$$\eta' = \eta - (\eta_1' + \eta_2' + \eta_3' + \eta_4') \quad (26)$$

$$\beta = \arcsin[\operatorname{sech}(\eta') \cdot \sin(\xi')] \quad (27)$$

$$l = \arcsin\left[\frac{\tanh(\eta')}{\cos(\beta)}\right] \quad (28)$$

$$Q = \operatorname{arsinh}[\tan(\beta)] \quad (29)$$

$$Q' = Q + e \cdot \operatorname{artanh}[e \cdot \tanh(Q)] \quad (30)$$

$$Q' = Q + e \cdot \operatorname{artanh}[e \cdot \tanh(Q')] \quad \text{iterointi, kunnes muutos} = 0 \quad (31)$$

$$\varphi = \arctan[\sinh(Q')] \quad (32)$$

$$\lambda = \lambda_0 + l \quad (33)$$

Käytännössä kaavassa (31) riittää kolme iteraatiokierrosta. Ensin kaavalla (29) laskettu Q sijoitetaan kaavaan (30), jolla saadaan Q' :n ensimmäinen likiarvo. Q' :n likiarvo sijoitetaan kaavaan (31), jota iteroidaan.

Meridiaanikonvergenssi

Syöte: pisteen geodeettiset koordinaatit (φ, λ), keskimeridiaani λ_0
Tulos: meridiaanikonvergenssikulma (γ)

$$\gamma = l \cdot \sin(\varphi) \cdot \left[1 + \frac{1}{3} V^2 (2V^2 - 1) \cos^2(\varphi) \cdot l^2\right], \quad \text{missä } l = \lambda - \lambda_0 \quad (34)$$

Mittakaavakorjaus ja pituuskorjaus

Syöte: pisteen geodeettiset koordinaatit (φ, λ), keskimeridiaani λ_0
 Tulos: mittakaavakerroin (k) pisteessä (φ, λ)

$$k = k_0 \left[1 + \frac{1}{2} \cos^2(\varphi) \cdot l^2 \right], \quad \text{missä } l = \lambda - \lambda_0 \quad (35)$$

Mittakaavakerroin muuttuu hitaasti ja sitä voidaan monissa tarkoituksissa pitää vakiona noin 10 km² :n suuruisella alueella. Pidemmällä etäisyyksillä mittakaavakerroin voidaan laskea kaavalla:

$$k = 1/6(k_1 + 4k_m + k_2) \quad (36)$$

missä k_1 ja k_2 ovat mittakaavakertoimet linjan päissä ja k_m on mittakaavakerroin linjan puolessavälissä.

Maastomittauksissa voidaan käyttää kaavalla (35) määritettyä mittakaavakertoimen arvoa k , mikäli esimerkiksi takymetrissä tehdään muutkin etäisyyksimittauksen reduktiot mittaushetkellä. Muuten ellipsoidin pinnalla tehty ja ellipsoidin kaareksi redukoitu etäisyyshavainto korjataan projektiotasolle kertomalla se k :lla, joka saadaan joko kaavalla (36) tai tasokoordinaateista laskien kaavalla:

$$k = k_0 \left[1 + \frac{1}{6R^2} (E_1^2 + E_1 E_2 + E_2^2) \right] \quad (37)$$

Suuntakorjaus

Ellipsoidin pinnalla havaittu suunta korjataan tasolle suuntakorjauksella (Arc-to-chord Correction $\delta = T - t$). Suuntakorjaus δ lasketaan seuraavilla kaavoilla:

Syöte: kahden pisteen koordinaatit (N_1, E_1) ja (N_2, E_2)
 Tulos: suuntakorjaus δ_{1-2} ja δ_{2-1}

$$\delta_{1-2} = \frac{1}{6R^2} (N_2 - N_1)(2E_1 + E_2) \quad (38)$$

$$\delta_{2-1} = \frac{1}{6R^2} (N_1 - N_2)(2E_2 + E_1) \quad (39)$$

Kaavoissa (37) – (39) käytettävistä itäkoordinaateista E_1 ja E_2 on vähennettävä 500 000 m (itäkoordinaatin arvo keskimeridiaanilla). R on keskikaarevuussäde, joka voidaan laskea kaavalla:

$$R = \sqrt{M \cdot N} \quad (40)$$

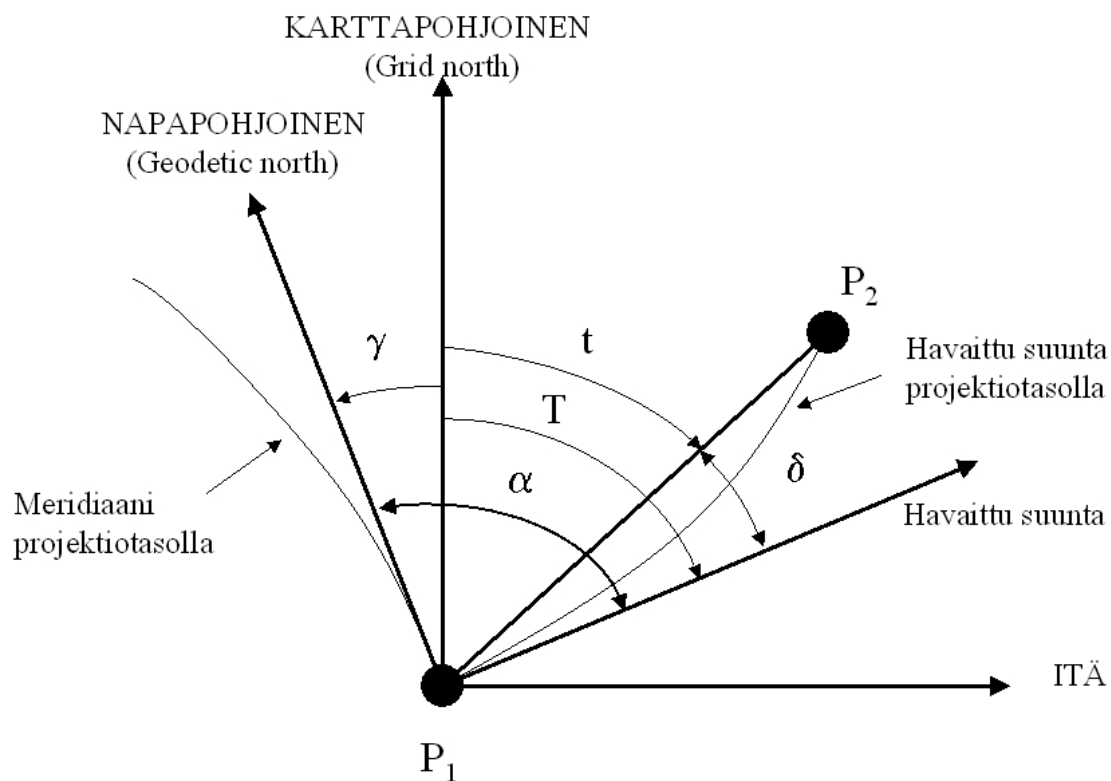
$$M = \frac{c}{V^3} \quad (41)$$

$$N = \frac{c}{V} \quad (42)$$

$$c = \frac{a^2}{b} \quad (43)$$

Suuntakulma (t) projektiotasolla, atsimuutti (α), konvergenssikulma (γ) ja suuntakorjaus (δ) riippuvat toisistaan seuraavan kaavan mukaisesti:

$$t = \alpha - \gamma - \delta \quad (44)$$



Kuva 4. Karttapohjoinen, napapohjoinen, konvergenssikulma ja suuntakorjaus.

Kun piste on keskimeridiaanin itäpuolella, karttapohjoinen on itään napapohjoisesta ja meridiaanikonvergenssi on positiivinen. Piste on ollessa länteen keskimeridiaanista, karttapohjoinen on länteen napapohjoisesta ja meridiaanikonvergenssi on negatiivinen. Katso myös kuva 3.



TECHNICAL DATA SHEET

ROBIN 3D FLEX

Horizontal scanning S-band Radar

Weight	133 kg including antenna
Beam	Width 1.8 degrees, 25 degrees
Speed	45 RPM for improved tracking capabilities
Range	Over 10 kilometers for large birds and flocks
Operation	Provides latitude and longitude of birds



3D tracking FMCW X-band Radar

Weight	65 kg including antenna
Beam	Width 0.95 degrees, 20 degrees height
Speed	30 RPM in elevation and 15 RPM in azimuth
Range	Effective ranges up to 3.5 to 5 kilometers
Operation	Supports staring mode for wing beat measurement, 2-axis scanning mode for 3D coverage and tracking mode with either manual or automatic track selection

Bird Data Acquisition System

Units	Radar Processing units for acquisition of raw signals Registration Systems with 16 bit/120MHz ADC Bird Monitoring System object detection and multi-hypothesis tracking software MySQL database free and open source relational database management system Shock proof SKB boxes for easy handling
Handling	
Access	Rack mounted console for direct access, network switch for remote access
Power	220V/10A including UPS
Software	Robin Visualizer real-time display, observation support, user defined overlays, alarms and playback Robin Report Viewer web based graphical view of database data and export of KML Robin Remote Monitor for remote control, fault detection and diagnosis of vital hardware and software processes
Operating system	Linux Ubuntu LTS free and open source providing optimum stability.
Additional	Weather station unit with integrated GPS providing location, weather data including wind speed and direction. Used for high wind speed protection of the radars and as an optional time synchronization method Ruggedized Laptop Wireless and GPS integrated laptop shipped with the Robin Visualizer including the Robin Observation Module for fast logging of field observations.

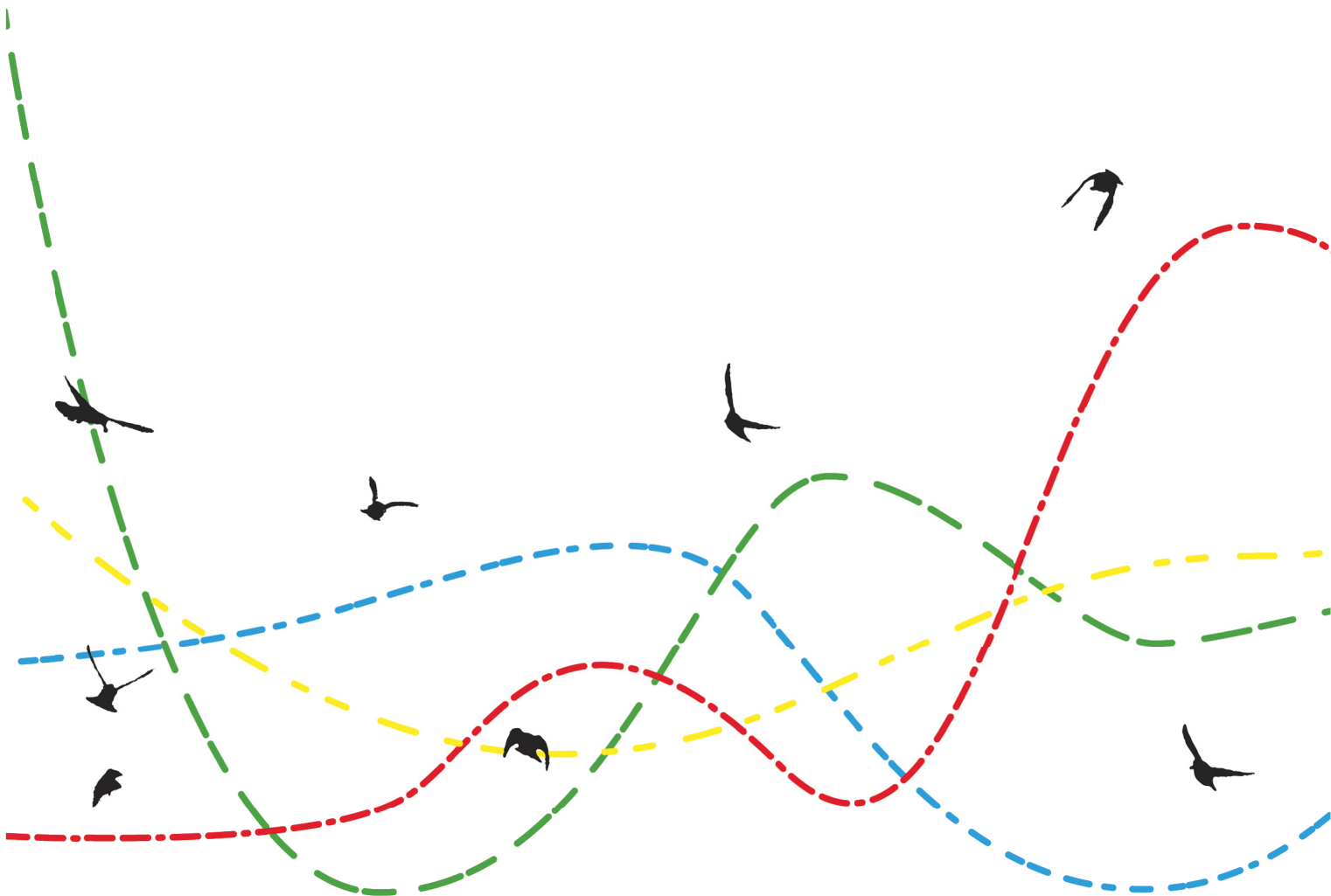


Advanced, Accurate, Adaptable

Robin Radar Systems BV, Mercuriusweg 1A, 2516 AW The Hague, the Netherlands

ROBIN Specification Guide

*Version 1.5
June 2017
Robin Radar Systems B.V.*



History description

Version	Date	Author	Comments
1.1	31 March 2015	René Somer	Updated
1.2	April 2015	René Somer	Updated
1.3	July 2015	René Somer	Updated
1.4	April 2017	Gerben Pakkert	Updated
1.5	June 2017	Gerben Pakkert	Updated

Disclaimer

Buyer and/or User acknowledge and agree that the ROBIN system(s) is/are innovative products subject to on-going further development. No guarantee, express or implied, is made as to the ability of any ROBIN system to monitor all bird traffic. ROBIN accepts no liability for any damage suffered by Buyer and/or User as a result of missing or incorrect data from the ROBIN system(s)

ROBIN Radar Systems B.V.

Horizontal S-band radar:

	Vendor	Furuno Electric Co. Ltd
	Antenna	
	Type	SN36AF
	Dimensions	3765x180x480 mm
	Weight	40 kg
	Angle	1.8/25 degrees
	Gain	26,8 dB
	Transceiver / receiver	
	Frequency	S-band (3050 MHz)
	Bandwidth	+/-30 Mhz
	Output power	60 kWatt
	Pulse length	S (80 nano sec)
	PRF	1900 Hz
	Range	10 km (2 SAT)
	Instrumented range	10 km
	Number of blanking sectors	2
	Range accuracy	0,5 meter
	Angular accuracy	0,035 degrees
	Range cell resolution	5 meter
	Angular cell resolution	0,35 degrees
	Range spacial resolution	30 meter
	Angular spacial resolution	1.8 degrees
	Motor Unit	
	Type	RSB111 60kW
	RPM	45
	Weight	93 kg
	Used outside Materials	
	Antenna	Fibre-reinforced plastic (FRP)
	Motor Unit	Steel
	Cable	PUR-black

ROBIN Specification Guide – For informational purposes only.

X-band radar:

	Vendor	Furuno Electric Co. Ltd
	Antenna	
	Type	XN24AF
	Dimensions	2550x140x255 mm
	Weight	13 kg
	Angle	0,95/20 degrees
	Gain	31,5 dB
	Transceiver / receiver	
	Frequency	X-band (9410 MHz)
	Bandwidth	+/- 30 Mhz
	Output power	25 kWatt
	Pulse length	S2 (150 nano sec)
	PRF	3000 Hz
	Range	5 km (2 SAT)
	Instrumented range	6 km
	Number of blanking sectors	2
	Range accuracy	0,25 meter
	Angular accuracy	0,018 degrees
	Range cell resolution	2,5 meter
	Angular cell resolution	0,18 degrees
	Range spacial resolution	30 meter
	Angular spacial resolution	0,95 degrees
	Motor Unit	
	Type	RSB-096 25kW
	RPM	24
	Weight	42 kg
	Used outside Materials	
	Antenna	Fibre-reinforced plastic (FRP)
	Motor Unit	Steel
	Cable	PUR-black